AUTOMATED 1-AXIS DRILL

Members:

Mary Mbaire Mbugua	**E022-01-0745/2019**
Renox Kipkoech	**E022-01-0790/2019**
Allan Kyalo	**E022-01-1999/2019**
Stephen Gitau	**E022-01-1298/2013**
Morris Lesinko	**E022-01-2298/2019**
Aswa Philip Katiechi	**E022-01-0778/2019**

BSc. Mechatronics Engineering

A Research Thesis Submitted in Fulfillment for the Award of the Degree of Bachelor of Science in Mechatronics Engineering in the School of Engineering, Dedan Kimathi University of Technology 2022/2023

Declaration

This research thesis is our original work and has not been presented in any university/ institution for a degree or for consideration of any certification.

Mary Mbaire Mbugua
DeKUT Reg. No: E022-01-0745/2019

Signature..... Date.....

Renox Kipkoech
DeKUT Reg. No: E022-01-0790/2019

Signature..... Date.....

Allan Kyalo
DeKUT Reg. No: E022-01-1999/2019

Signature..... Date.....

Stephen Gitau
DeKUT Reg. No: E022-01-1298/2013

Signature..... Date.....

Morris Lesinko
DeKUT Reg. No: E022-01-2298/2019

Signature..... Date.....

Aswa Philip Katiechi
DeKUT Reg. No: E022-01-0778/2019

Signature..... Date.....

This research thesis has been submitted for examination with our approval as University Supervisors.

Dr. Titus Mulembo
Department of Mechatronic Engineering

Dedan Kimathi University of Technology (DeKUT), Kenya

Signature..... Date.....

Acknowledgement

We wish to express our thanks to the almighty God for His love, protection and good health in our life. We also appreciate our entire families for their financial and emotional support throughout our education. Special commendation goes to our supervisor, Dr. Titus Mulembo for his guidance and support all through this research work. Special commendation also to our colleagues their special guidance in this work and always responding to my many questions. We would also like to mention his continuous words of encouragement and support and always believing in us and pushing us to be the best we can ever be. Our final appreciation goes to Dedan Kimathi University of Technology for providing us with an opportunity to pursue our Bachelor of Science studies.

Table of Contents

List of Figures

List of Tables

Abbreviations

CAD: Computer-Aided Design

DOE Design of Experiments

DVA: Dynamic Vibration Absorber

FEA Finite Element Analysis

FEM: Finite Element Method

MSE Mean square of errors

NVH: Noise, Vibration

OAD: One-Axis Drill

PID: Proportional-Integral-Derivative

RMS Root mean square errors

SDM: Structural Dynamics Modeling

SSE Sum of squared errors

SVS: Structural Vibration Suppression

VM: Vibration Mitigation

Nomenclature

Vibration Related Nomenclature

ω: Angular frequency.

A: Amplitude of vibration.

f: Frequency of vibration.

x, y, z: Displacement in the x, y, and z directions, respectively.

v: Velocity.

a: Acceleration.

T: Period of vibration.

ζ: Damping ratio.

Δx, Δy, Δz: Differential displacements in x, y, and z directions.

ν: Poisson's ratio.

κ: Spring constant.

φ: Angle or phase

T: Torque

SNR: Signal-to-Noise Ratio.

∇ 3x1 vector of partial derivative operators

Statistics Related Nomenclature:

x: Mean.

σ: Standard deviation.

ρ: Correlation coefficient.

X, Y, Z: Random variables.

N: Sample size.

n: Number of observations.

α: Significance level.

β: Type II error.

ANOVA: Analysis of Variance.

CI: Confidence Interval.

CDF: Cumulative Distribution Function.

PDF: Probability Density Function.

H0, H1: Null and alternative hypotheses.

ε: Error term in statistical models.

Abstract

High-quality drilled holes are required in a lot of industrial and medical applications. It is imperative to design a low-cost and precise drilling machine with real time monitoring of vibrations, temperature and forces applied to the workpiece during the drilling operation. This project focuses on the development of a one-axis drill system utilizing a polymer workpiece for improving bone drilling operations in medical applications.

Chapter 1: Introduction

1.1 Background

Over the years, drilling procedures have come a long way as technology has continued to develop more accurate and efficient machines. Traditionally, the drilling operations were carried out manually. Perforations were made by means of hand-held lithic borers which preceded clockwise/counterclockwise rotational drilling. This method was effective on soft stone but ineffective on harder material (A. John Gwinnet 1998).

Technology has helped in transformation of the drilling processes into automatic process with real time monitoring and correction during errors. For example, bone drilling in medical environment is one field where the drilling process is used and has to be done precisely.

PMMA (polymethyl methacrylate) is a biomimetic material. This renders it appropriate for uses that need a tough and long-lasting material. It has high strength, fatigue resistance and good dimensional stability and it has many applications used in aerospace, automotive or medical devices industry. PMMA's near perfect resemblance with bone is due to its very high strength and fatigue resistance characteristics which are the hallmarks of any tissue including bones. Moreover, PMMA possesses high dimensional stability implying its ability to retain the shape and size regardless of stress introduced therefore recommended for use in applications that demand resilience.

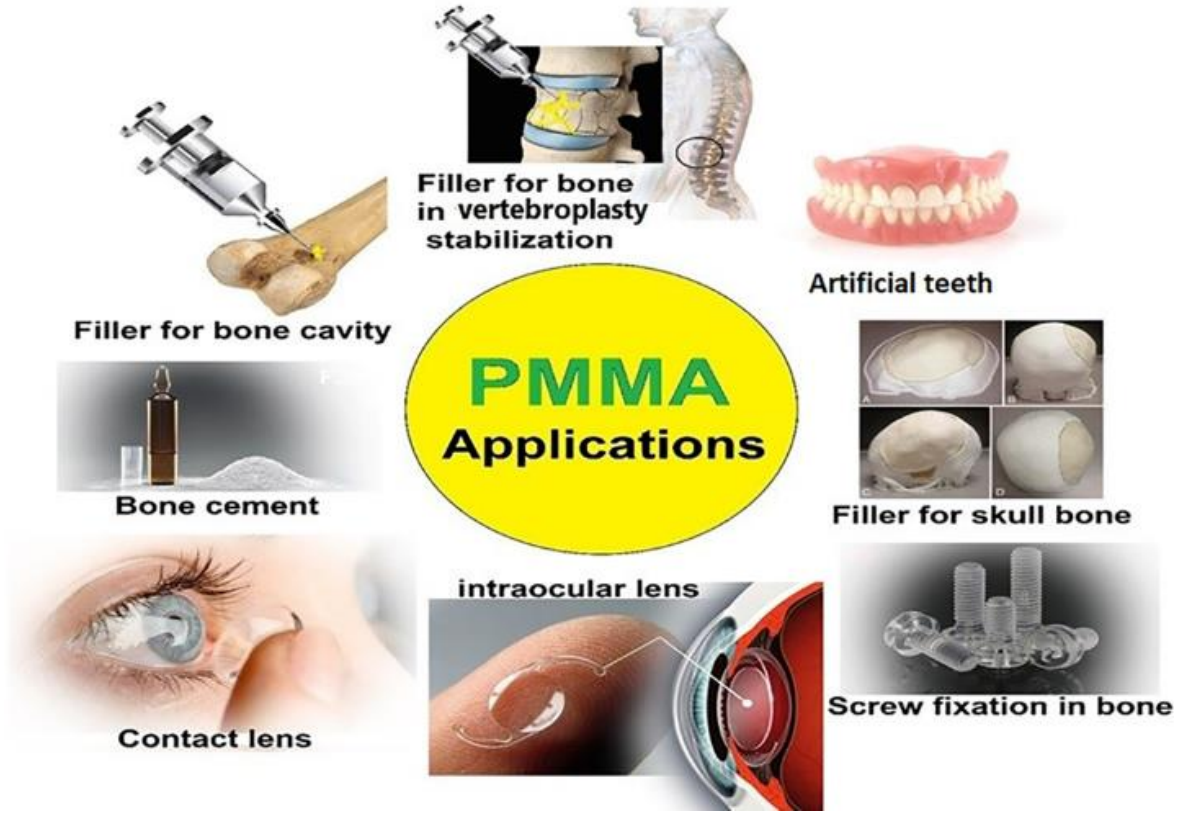


Figure 1: PMMA applications (http://tinyurl.com/4ucw3s7y)

The development of automated one-axis drill for drilling PMMA material, which has properties similar to bone, is a promising area of research. The benefits of using automated drilling systems for PMMA material include improved precision, reduced risk of human error, efficiency, and safety in the drilling process.

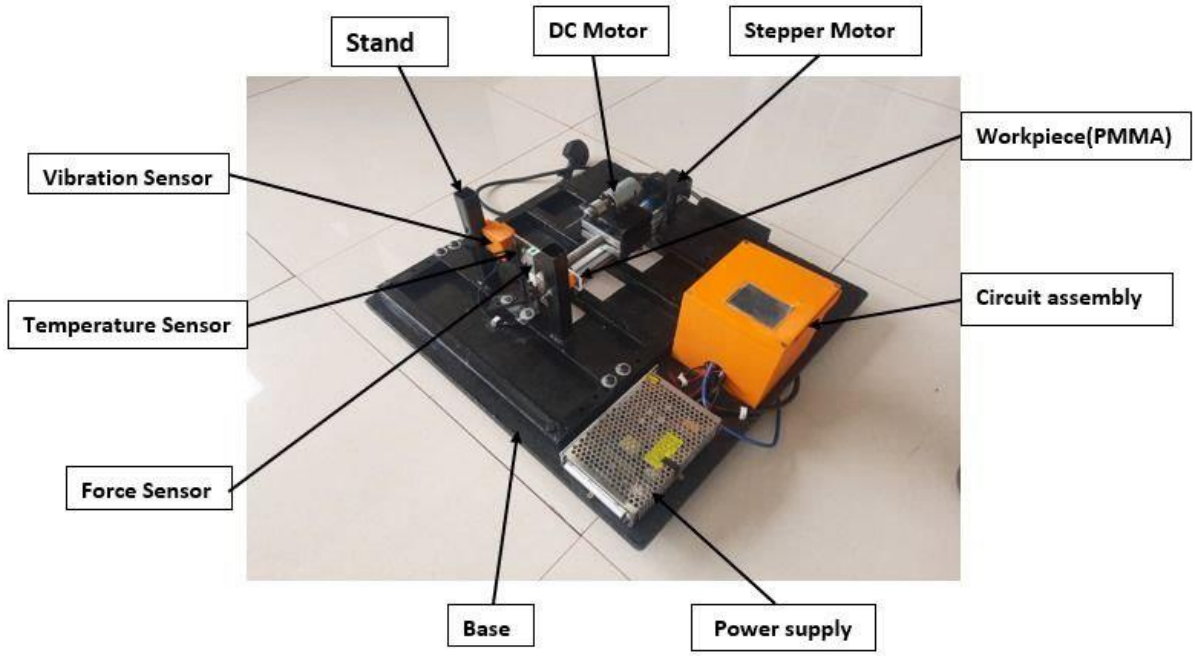


Figure 2: An automated one-axis drill machine

1.2 Problem statement

In various industrial applications, high-quality drilled holes are essential. The current industrial drilling machines are expensive and lack the required precision for achieving high-quality drilled holes. There is therefore need to address the challenges encountered in drilling processes such as workpiece damage, chatter, unmonitored and unregulated vibrations, drill bit breakage, and misaligned holes.

A study and analysis of the drilling parameters, including vibrations, force, drill speed, and temperature is to be done in order to have a better understanding of their effects on the drilling process. These parameters directly affect the quality of the resulting drilled hole. An automated system that can control these parameters to avoid poor drilled-hole quality can the then be developed from the experiments, study conducted and analyzed data. The system will be designed to monitor and regulate vibrations, force, drill speed, and temperature during the drilling process, ensuring that the drilling parameters are within the optimal range.

1.3 Objectives

1.3.1 Main objective

- Development and fabrication of a precise automated one axis drill for PMMA(polymethyl methacrylate) drilling.

1.3.2 Specific objectives

- Carry out experiments to determine best drilling parameters
- Carry out statistical analysis of actual drill parameters.
- Vibration mitigation for high quality drilled holes.
- Run physical model of automated drill.
- Monitor drilling parameters in real time.

1.4 Significance of the study

With continued advancements in technology, there is a great need to come up with more superior ways to overcome challenges faced in the biomedical field through engineering. Drilling of bones being an invasive procedure has to be done in a manner that would result in minimum-to-no harm to the patients. PMMA being a material that is affordable and has properties similar to that of bone offers a great chance to delve into the study of the drilling process and develop an affordable, precise and automated one-axis drilling machine. The challenges encountered in the drilling processes can be studied and a conclusive solution can be formulated on the best parameters for performing the drilling processes. Based on these findings, an automated one-axis drill capable of self-correction can be developed and this can be translated to the biomedical field and also used in other fields such as aerospace.

1.5 Scope of work

The study encompasses the investigation of the challenges faced during PMMA drilling processes; specifically, vibrations, load, drill speeds and temperature, analysis of data obtained from the investigations and determination of optimal drilling parameters. A working automated one-axis drill is also to be developed which will be used in the study.

Chapter 2: Literature Review

2.1 Introduction to literature review.

The purpose of the literature review is to trace automatization in the development of one-axis drill technology. It will concentrate on the achievements and currently available literature in this field aiming at identifying what has been done, where it still needs corrections or further work to perfect automated one-axis drill technology.

The resulting research helps to understand different aspects associated with automated drilling technology that include development of CNC gantry machine concepts, the model for automating 3D CAD "drill modeling software", and benefits regarding use of it. Further, there are the papers describing how to develop accurate models of conical twist drills and positional synthesis of variation for automated systems.

These resources together contribute to the knowledge about automated drilling technology and its applications, such that a broad base is laid down for literature review concerning development of an automatic one-axis drill.

2.2 Evolution of Automated drilling technology

The first drilling automation can be seen in the early manual approaches, which represented another step to shift away from hand-dug wells into mechanized drilling. This process developed through a number of stages from the spring-pole technique, into percussion drilling or cabling and ultimately via rotary drilling with pipe tools. (Alfred, 2007)

In the late 20 th century computer- controlled systems started their rise which is needed to combine real time data with mechanized equipment so that drilling rate of penetration (ROP) could be improved and wellbore stability can also be maintained. One of the first cases to illustrate this type of integration was rig-automated drilling in 1990's (2015)

Over the 21st century, there have been significant developments in automated drilling systems. 2014, for instance, ConocoPhillips and National Oilwell Varco NOV finalized a pilot program in Texas to test a new automated system that significantly reduced drilling time by over 40%. This technology was later extended to North Dakota and the North Sea (Catalin, 2021)

Another notable development was the launching of a DrillTronics program by Sekal, Norwegian software firm that could regulate torque, pump pressure and hook load on an offshore rig owned by Statoil (Anon. 2015).

Contemporary drilling automation landscape is known for augmenting productivity and quality, enhancing personnel safety while managing risk in an effective manner. Some of the main areas that affect include well complexity, data overload, efficiency repetitive well manufacturing and HSE issues. (Anon 2015.)

Although it has faced some resistances and challenges, parts of the drilling industry have full achieved a level where automation is made on commercial basis as there has been considerable collaboration among different parties to come up with standardized room for data flow line automation process (Anon., 2015) Automation in drilling technology has a long history of constant invention and adaption. Future advancements in automated drilling, especially in terms of precision, safety, and environmental effect, promise to further transform the industry as technology continues to progress.

2.3 Latest advancements in Automated drilling technology.

Present-day automated drilling technology therefore mirrors a highly sophisticated amalgamation of complex computer systems, mechanized machinery and inventive drilling procedures. The latest developments in this field have been the recent studies and industry reports.

Today's automated drilling systems are very dependent on computing power to increase efficiency and accuracy. For example, the combination of real-time data analytics with automated equipment has greatly accelerated drilling pace and wellbore stability. Companies such as ConocoPhillips and National Oilwell Varco have also shown that these systems could work when they recorded massive reductions in drilling time, increase in operational efficiency through successful pilot programs conducted (Cayeux et al., 2021).

Also, recent innovations have been observed in drill bit design and software for drilling. For instance, Sekal's DrillTronics program demonstrates how software can enhance drilling operations through control of essential parameters like torque, pump pressure and hook load. One of them includes this drilling program which represents a shift towards more intelligent and responsive systems in the petroleum industry (Cayeux et al., 2021).

Automated drilling technology has also been applied for use in even more difficult environments such as operations offshore platforms and unconventional reservoirs where precision and safety are most important. The automation of processes in these environments does not only increase efficiency but also lifts worker safety by reducing human involvement into hazardous operations (Lauren, 2023).

The development of automated drilling technology has been a gradual process where each step towards more advanced automation made the oil well and gas drilling operations run in an even better, safer and cleaner way. Automated drilling is on the verge of an even greater revolution as technology continues to advance.

2.4 Materials and testing in automated drilling technology (PMMA)

The original acrylic PMMA, poly (methyl methacrate) has many useful traits that make it a superior alternative to glass in numerous instances. PMMA has become a preferred material in various industries due to its transparency, shatter resistance and UV stability (Anon 2022).

PMMA can be molded in various ways, and it is easy to bind itself with other materials for enhancement of the properties as well. Thermoforming is flexible when heated and become solid if cooled. PMMA is also used in sound-proof rooms, audio studios and cars owing to its high surface hardness and resistance against abrasion

In terms of drilling, PMMA's transparency enables the analysis and monitoring of drilling mechanisms thus serves as a good material for experimenting or demonstrating. The utilization of PMMA in drilling experiments especially using laser drilling procedures has been widely investigated. Studies have revealed that it is possible to drill PMMA efficiently with CO 2 lasers, and the material helps in understanding various aspects of the entire drilling process such as how different parameters like laser power supply, exposure time etc. impacts on its success. (Abeer A et al., 2020)

Machining PMMA poses some unique challenges for instance, one must accurately control cutting parameters so not to end up with the following problems potion swarf evacuation; reattachment of cut material, melting and even distortion in part may take place. The final quality of the drilled holes is also significantly determined by cut and geometry of a killer. Balancing spindle RPM, feed rate and cutter geometry are essential to drill PMMA in order to obtain the desired outcomes. So, these parameters need to be carefully adjusted so that it matches with the shear requirements of material and also, defects like burr formation cannot take place as an after effect due to excessive heat along with inadequate swarf extraction (Abeer A et al., 2020).

Understanding these properties and the PMMA's behavior under various drilling conditions helps in improving the drilling techniques and developing new applications where precision and the material's integrity are most crucial. The PMMA's role in drilling technology can be as a test material for experimental setups and also a subject for understanding the impact drilling parameters on the material behavior.

2.5 Challenges and the limitations in Automated Drilling Technology

Automated drilling technology suffers from several challenges and limitations;

2.5.1 Technical Challenges in Automation

Complexity in System Integration: It takes complicated and costing software programs and hardware solutions to integrate various subsystems such as sensors, control systems, mechanical components in the architecture.

Reliability and Maintenance: Normally, keeping elaborate systems up and running in demanding drilling settings while meeting stringent standards of reliability proves to be a daunting task; there is always the specter of system failure followed by an expensive stoppage.

2.5.2 Data Management and Analysis:

Handling Large Volumes of Data: Automated drilling produces large volumes of data that need to be processed and analyzed almost immediately in real-time, making the management process a complex operation when it comes to managing this huge quantum of information and obtaining actionable predictive insights.

Cybersecurity Concerns: In the modern approach to automated drilling systems, increased digitalization makes it vulnerable to attacks by cyber threats Thus a focus should be optical regarding data security and control system integrity.

2.5.3 Operational Challenges:

Adaptability to Different Geological Conditions: It is challenging to design automated systems that can adjust themselves according to different geological conditions because such designs have also needed their versatility.

Human-Machine Interaction: Balancing between automation and human oversight to ensure both safety and efficiency is no easy task.

2.5.4 Economic and Environmental Challenges:

High Initial Investment: Establishing and installing the automated drilling technology is an expensive venture especially for small companies.

Environmental Concerns: Although automation can provide better efficiency and lessen impact on the environment, making these useful components calls for trouble with nature.

2.5.5 Regulatory and Standardization Issues:

Lack of Standardized Regulations: The lack of standardized regulations, which are specific to automated drilling technology can also pose a challenge for the industry in this direction.

Compliance with Safety Standards: Making sure that the automated systems meet current safety standards and regulations, especially in international operations where such standard differ is not easy but essential.

2.5.6 Technological Limitations:

Sensor Accuracy and Limitations: This in automated drilling can affect the total efficiency and safety of a system, which also stresses on more precise as well as reliable sensors.

Software limitations: The existing software might not fully account for all factors, in drilling operation hence leading to limitations in the automation capabilities.

2.6 Emerging Trends in Drilling Automation

2.6.1 AI Integration and Advanced Control Systems

The coupling of AI and ML technologies in control systems is an important trend. These technologies facilitate predictive analytics, real-time decision making and increased drilling accuracy. Artificial Intelligence in drilling automation is transforming how drilling data gets processed and used for enhancement of operations.

Enhanced Data Analytics and IoT

Drilling operations are more and more monitored through the Internet of Things (IoT) advanced data analytics. Once collected, data from sensors and IoT devices are analyzed to develop designs that improve drilling methods for efficiency and safety.

2.6.2 Advancements in Material Science

High-Performance Drilling Materials: More advanced materials for drilling equipment is the focus of research in material science, which includes high-performance alloys and composites that can withstand extreme environments when it comes to drilling.

Smart Materials: Research on smart materials that can transform their properties when they sense changes in the environment is increasingly happening. These materials could substantially improve the adaptability and productivity of drilling activities.

2.6.3 Safety and Environmental Sustainability Improvements

Automated Safety Systems: Automation of the safety systems that are meant to predict and prevent accidents is a developing area in research which seeks at eliminating human error thus enhancing overall safety in drilling operations.

Environmentally Friendly Drilling Practices: There is a growing emphasis on creating technologies for drilling with minimal environmental consequences, including research in fuel efficiency, reduced emissions and techniques that affect the ecological footprint of these activities.

2.6.4 Human-Machine Interaction and Workforce Development

Ergonomics and Interface Design: Currently, research is concentrating on the ergonomics of human-machine interfaces to design systems that are intuitive and improve collaboration between humans and automated systems.

Training and Skill Development: As automation shifts, there is a growing demand for workforce training and development of skills as well as research in the most effective ways to train drilling professionals about this changing technology landscape.

2.6.5 Standardization and Regulatory Development

Development of Industry Standards: One of the main subjects being researched is also standardization in automated drilling technology, as it requires wider acceptance and compatibility between various operations.

Regulatory Frameworks: Keeping up with technological advancements by adapting the regulatory frameworks to ensure that automated drilling technologies comply with current regulations and identifying new challenges arising from automation is an area of substantial research.

Automated drilling technology continues to evolve Incorporating sophisticated control systems advanced data analytics material science advances the focus is on safety environmental sustainability considerations of human-machine interaction workforce development and regulatory frameworks. These trends indicate continuous efforts to overcome challenges as well as drive continuing innovation and adoption of automated drilling systems.

Chapter 3: Methodology

To accomplish the objectives, we had established for the project, we followed these steps to attain accomplishment.

3.1 Design and fabrication of PMMA stands

We intended to carry out experiments on different lengths of clamping the PMMA strips. The lengths were 5cm span, 10 cm and 15 cm span. The use of different spans of clamping the workpiece was meant to establish the effect of clamped distance on vibration, temperature distribution and load applied on the PMMA strip. We intended to conduct experiments for fast and slow drill speeds using the different spans to measure the variables stated.

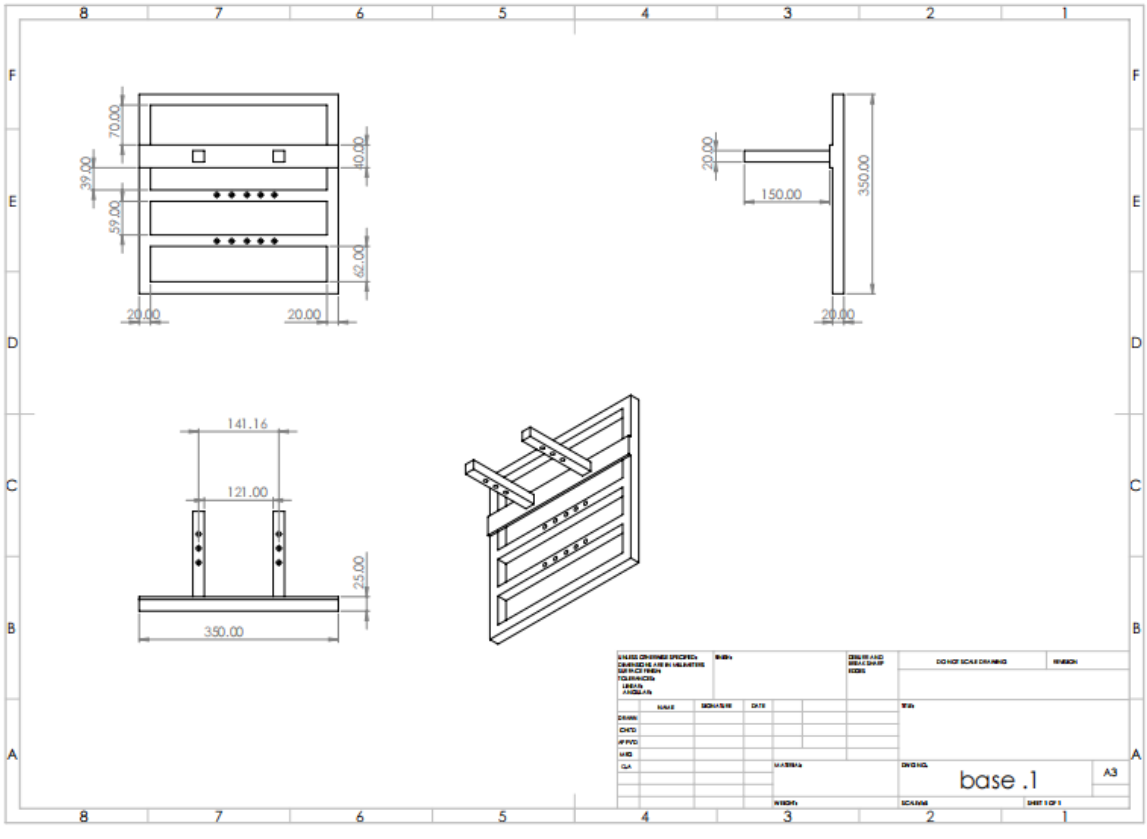


Figure 3: Schematic of one axis drill

A schematic of a magnetic manipulation system is as shown in Figure 3.

Figure 4: Photograph showing the replacements of the stands and the different spans we will use

3.2 Development of firmware in Arduino.

The functionality to be met was:

a) **Run and control the speeds of the feeding stepper motor and the drilling dc motor**

The dc motor was driven and controlled using a l298 driver using pulse width modulation signals to control the speed of drilling. The stepper motor was driven using a TB6600 driver. The l298 driver and tb6600 driver are powered from a 12V power supply to run the motors. A separate 5V signal was being sent from the microcontroller (Arduino Uno) to power and control the drivers and hence the motors. The dc motor in use is a 12V brushless motor which is attached to a drill bit via a chuck whereas the stepper motor is a NEMA23 stepper motor.

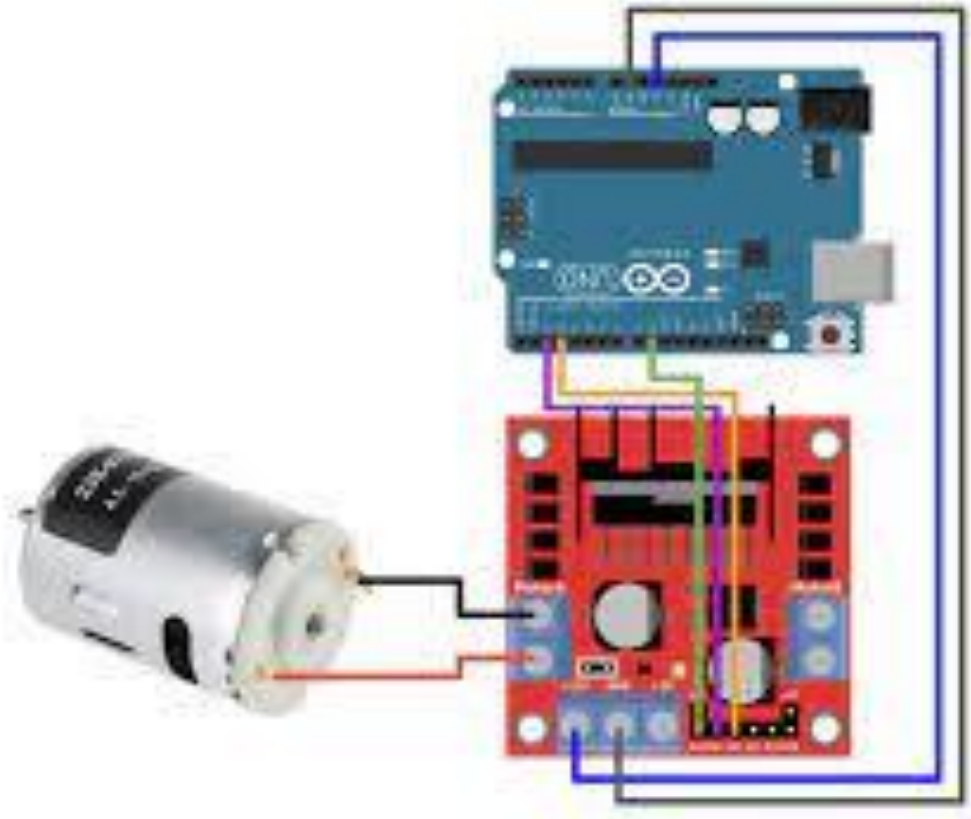


Figure 5: Circuit Design in Fritzn

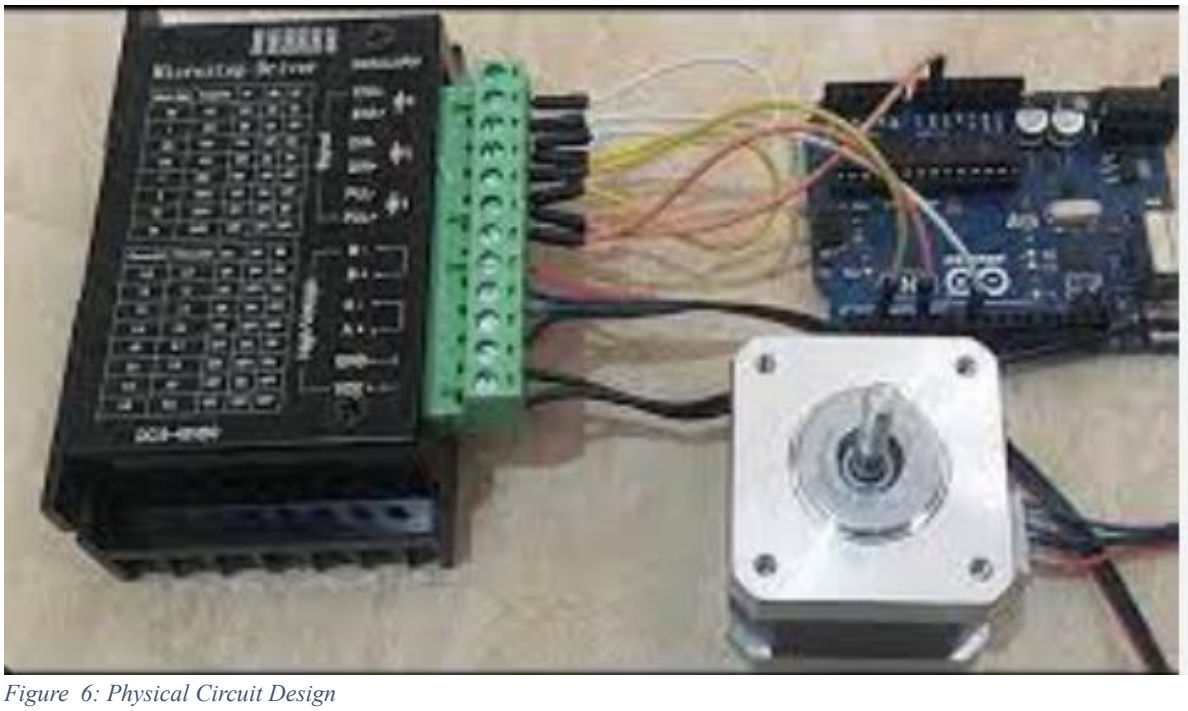


Figure 6: Physical Circuit Design

Code snippet to run the motors:

```
  pinMode(dirPin,OUTPUT);
  pinMode(enPin,OUTPUT);
  digitalWrite(enPin,LOW);
  // Turn off motors - Initial state
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
}

// Main loop
void loop() {
  runStepperMotor();
  runDCMotor();
}

// Function to run the stepper motor
void runStepperMotor() {
  // Set the stepper motor direction
  digitalWrite(dirPin,LOW);
 // Enables the motor to move in a particular direction
    digitalWrite(stepPin,HIGH);
    delayMicroseconds(16380);
    digitalWrite(stepPin,LOW);
    delayMicroseconds(16380);
}
// Function to run the DC motor
void runDCMotor() {
// Turn on motor A
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  analogWrite(enA, 255);
}

```

Figure 7:Code snippet for running motor

3.3 Reading the sensors

a) Temperature Measurement

The sensor in use were mlx90614 contactless temperature sensor. The features of MLX90614 that made it applicable for our application over other temperature sensors:

- Non-Contact Measurement: One of the main advantages of the MLX90614 is its ability to measure temperature without direct contact with the object. It uses infrared technology to detect the thermal radiation emitted by the object, making it suitable for measuring the temperature of moving objects, liquids, or hazardous materials.
- Wide Temperature Range: The MLX90614 is designed to measure a wide range of temperatures. It can handle both low and high-temperature applications, making it versatile for a variety of industrial and commercial uses.
- Accuracy: This sensor provides accurate temperature measurements. The integrated signal conditioning and digital output help maintain a high level of precision in temperature readings.
- Compact Size: The MLX90614 is compact and lightweight, making it easy to integrate into different systems and applications. Its small size is particularly beneficial in situations where space is limited.

- Low Power Consumption: The sensor is designed to operate with low power consumption, making it suitable for battery-powered devices and applications where energy efficiency is crucial.
- Digital Output: The MLX90614 provides a digital output, making it easy to interface with microcontrollers or other digital systems. This simplifies the integration process and enhances compatibility with various platforms.
- Configurability: The MLX90614 often comes with configurable parameters, allowing users to adjust settings such as emissivity, which is important for obtaining accurate readings from different surfaces.
- Fast Response Time: The sensor has a fast response time, allowing it to quickly capture changes in temperature. This feature is beneficial in applications where real-time temperature monitoring is essential.
- Durability: The MLX90614 is often built with durable materials, making it suitable for use in harsh environments or industrial settings where sensors may be exposed to challenging conditions.
- Cost-Effective: Compared to some other temperature sensing technologies, the MLX90614 is often considered a cost-effective solution for non-contact temperature measurement applications.

The mlx90614 sensor uses I2C protocol for communication with the microcontroller (Arduino Mega 2560) for data collection.

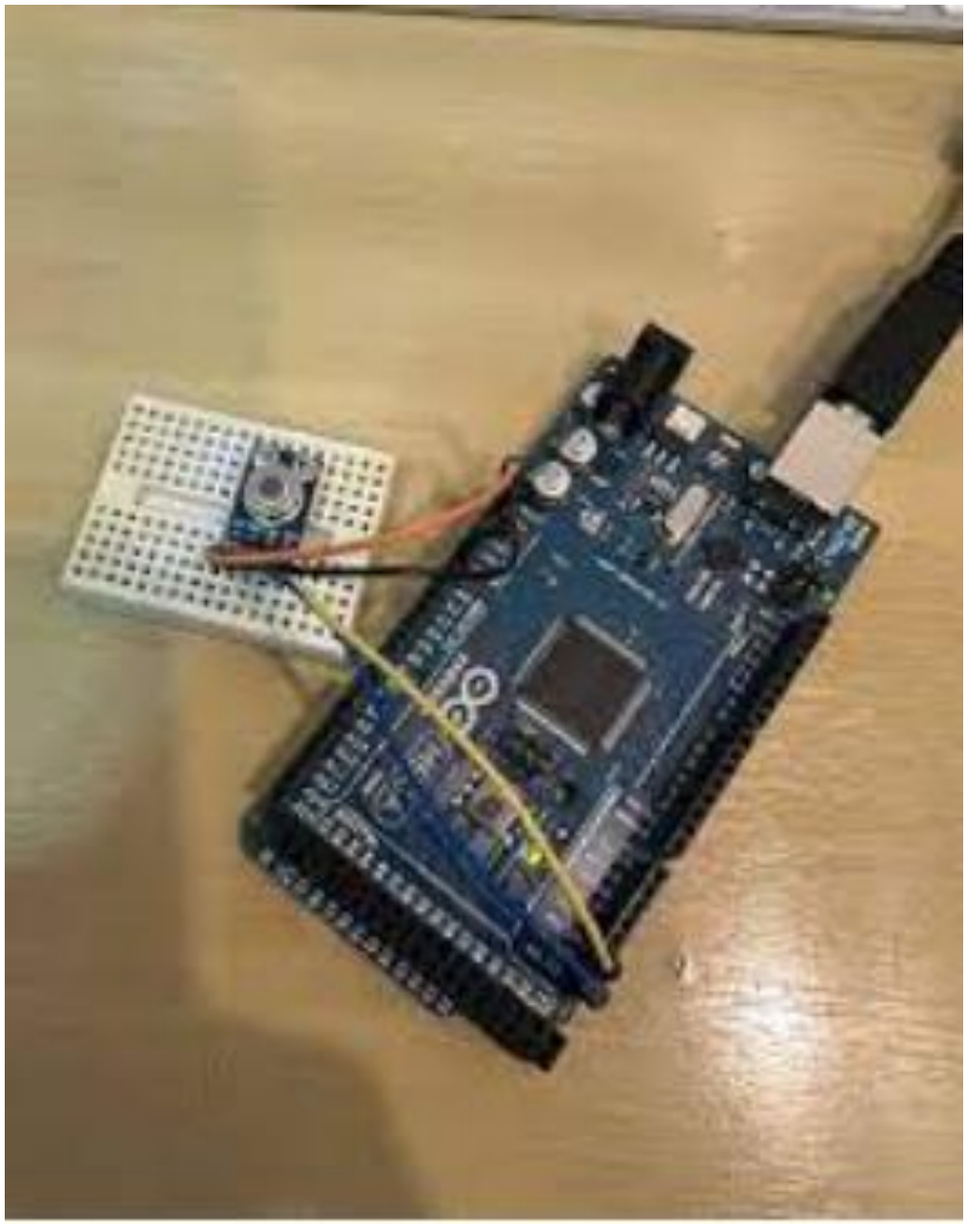


Figure 8: I2C Setup

Code snippet to read temperature values:

MLXANDARDUINO.ino

```ino
#include <Adafruit_MLX90614.h>

Adafruit_MLX90614 mlx = Adafruit_MLX90614();

void setup() {
  Serial.begin(9600);
  while (!Serial);

  if (!mlx.begin()) {
    Serial.println("Error connecting to MLX sensor. Check wiring.");
    while (1);
  };
}

void loop() {
  float objectTemperatureC = mlx.readObjectTempC(); // Read object temperature in Celsius

  // Send the object temperature in Celsius to the GUI
  Serial.print("C:");
  Serial.println(objectTemperatureC);

  delay(1000); // Delay to control the rate of data transmission
}
```

Figure 9: Code for reading temperature

b) Load measurement

The load applied to the PMMA strip during experimentation was measured using a load cell. The load cell was coupled with a hx711 driver that interprets and amplifies the signal to make it readable by the Arduino mega 2560 microcontroller. Digital pins were used to connect the driver the microcontroller to provide data and clock signals. The driver was powered with a 5V signal from the microcontroller. The load cell in use had a maximum load rating of 5kilograms. The load measured was a measure of deflection of the PMMA strip during drilling when clamped in different orientations. To be able to accurately measure the load using this sensor, calibration with a known load was required.

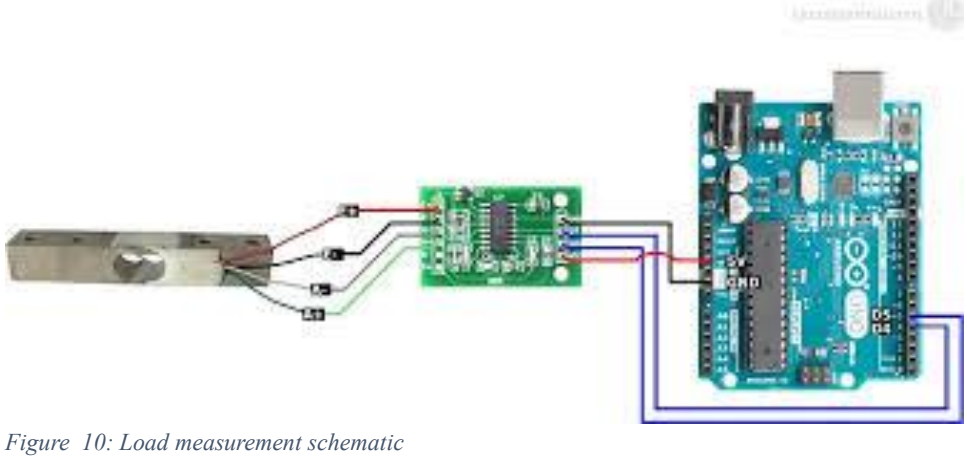


Figure 10: Load measurement schematic

Features of hx711 that motivated its use:

- 24-Bit ADC: The HX711 is equipped with a 24-bit analog-to-digital converter, providing high-resolution measurements for accurate weight readings.
- Selectable Gain: The driver allows users to select from multiple gain settings to amplify the small signals from load cells effectively. Common gain values include 128 and 64.

- Serial Interface: HX711 communicates with a microcontroller through a simple two-wire serial interface, which typically consists of a clock (CLK) and a data (DOUT) line. This makes it easy to interface with a variety of microcontrollers.
- Low Noise: The HX711 is designed to minimize noise and interference in the signal, ensuring accurate and stable measurements, especially in applications where precise weight readings are crucial.
- On-Chip Voltage Regulator: The built-in voltage regulator on the HX711 helps maintain stable operation even when the supply voltage fluctuates. This is beneficial for consistent and reliable performance.
- Tare Function: The driver supports a tare function, allowing the system to subtract the weight of containers or other fixed items from the overall measurement. This helps in obtaining net weight values.
- Selectable Data Rate: The HX711 driver often allows users to choose the data rate for conversion. Common data rates include 10 Hz and 80 Hz, allowing a trade-off between speed and power consumption.
- Zero Drift Calibration: To compensate for long-term drift in the load cell or other environmental factors, the HX711 driver supports zero drift calibration. This ensures that the zero point remains accurate over time.
- Low Power Consumption: The HX711 is designed with low power consumption in mind, making it suitable for battery-powered applications or situations where power efficiency is critical.
- Versatility: The HX711 driver is versatile and compatible with various load cells, making it a widely used solution in different applications, such as industrial scales, kitchen scales, and other precision weighing systems.

Code snippet to read load cell values:

LOADCELLANDARDUINOMAIN.ino

```
#include "HX711.h" //You must have this library in your arduino library folder
#define DOUT 9
#define CLK 8
HX711 scale(DOUT, CLK);
float calibration_factor = 96652;
void setup() {
  Serial.begin(9600);
  // put your setup code here, to run once:
  scale.set_scale(157860); //Calibration Factor obtained from first sketch
  scale.tare(); //Reset the scale to 0
}

void loop() {
  // put your main code here, to run repeatedly:
 Serial.print(scale.get_units(), 3); //Up to 3 decimal points
 Serial.println();
}
```

Figure 11: Code snippet to read load cell values:

3.4 Vibration measurement

To determine the extent of vibrations experienced during running and drilling process, an accelerometer was used; the particular model is ADXL345. The accelerometer measures acceleration along the x, y and z axes in units of m/s^2. The I2C protocol was used to establish communication between the sensor and

the microcontroller (Arduino mega 2560). The choice of an accelerometer instead of a digital vibration sensor was driven by the need to collect analog data of the variances experienced during drilling process; the z axis of the accelerometer was used for measurement; the changes of values of acceleration were used as the measure of acceleration. Other features of the accelerometer that made it suitable for application:

- High Sensitivity: The accelerometer has a selectable sensitivity range, allowing users to choose the sensitivity level that best suits their application. The higher sensitivity enables the detection of even small vibrations.
- Wide Dynamic Range: The ADXL345 has a wide dynamic range, making it capable of measuring both small and large accelerations. This is crucial for applications where the amplitude of vibrations may vary significantly.
- Low Noise and High Resolution: The device provides low noise levels and high resolution, ensuring accurate detection and measurement of vibrations, even in environments with minimal accelerations.
- Digital Output: The ADXL345 communicates with microcontrollers or other digital systems using I2C or SPI protocols, providing digital output for easy interfacing and integration with various platforms.
- Low Power Consumption: It is designed with low power consumption, making it suitable for battery-operated applications or situations where power efficiency is critical.
- Built-In Motion Detection and Activity/Inactivity Detection: The ADXL345 includes features for motion detection and activity/inactivity sensing. These features can be leveraged for triggering actions or alarms based on detected vibrations.
- User-Adjustable Bandwidth: The bandwidth of the accelerometer can be adjusted to filter out unwanted high-frequency noise, providing flexibility in adapting to specific vibration frequency ranges.
- Free-Fall Detection: The ADXL345 has a built-in free-fall detection function, which can be useful in applications where the sensor needs to detect the absence of vibration or a sudden drop.
- Compact Size: The small and compact form factor of the ADXL345 makes it easy to integrate into various devices and systems, allowing for flexibility in placement and application.

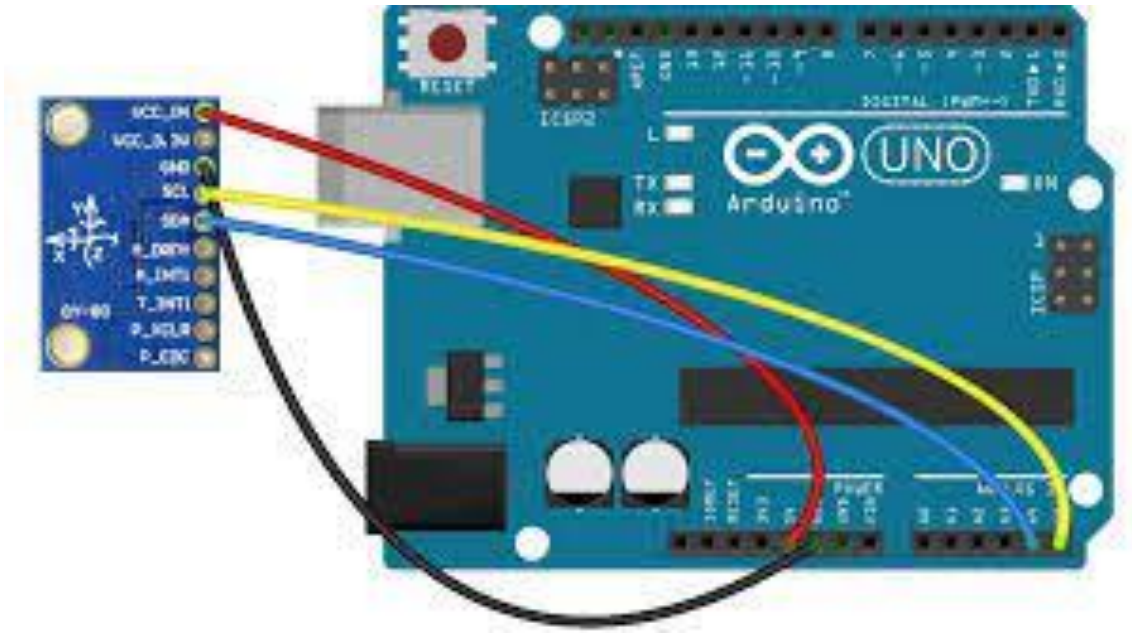


Figure 12: Vibration sensor setup

Code snippet to read acceleration values:

MLXANDARDUINO.ino

```
#include <Adafruit_MLX90614.h>

Adafruit_MLX90614 mlx = Adafruit_MLX90614();

void setup() {
  Serial.begin(9600);
  while (!Serial);

  if (!mlx.begin()) {
    Serial.println("Error connecting to MLX sensor. Check wiring.");
    while (1);
  };
}

void loop() {
  float objectTemperatureC = mlx.readObjectTempC(); // Read object temperature in Celsius

  // Send the object temperature in Celsius to the GUI
  Serial.print("C:");
  Serial.println(objectTemperatureC);

  delay(1000); // Delay to control the rate of data transmission
}
```

Figure 13: Code snippet to read vibration sensor values:

3.5 Project Assembly

The next step involved bringing the individual tested sections of the project together to develop the prototype. The project comprises of two Arduinos; an Arduino Uno is used to run and control the motor speeds and an Arduino Mega 2560 that reads the different sensor values. The microcontrollers and the drivers were placed within one enclosure to keep the circuitry neat. The sensors were positioned in the best possible places closest to the point of measurement to enable accurate data collection. Other components of project assembly included repainting of the setup and wiring of components.

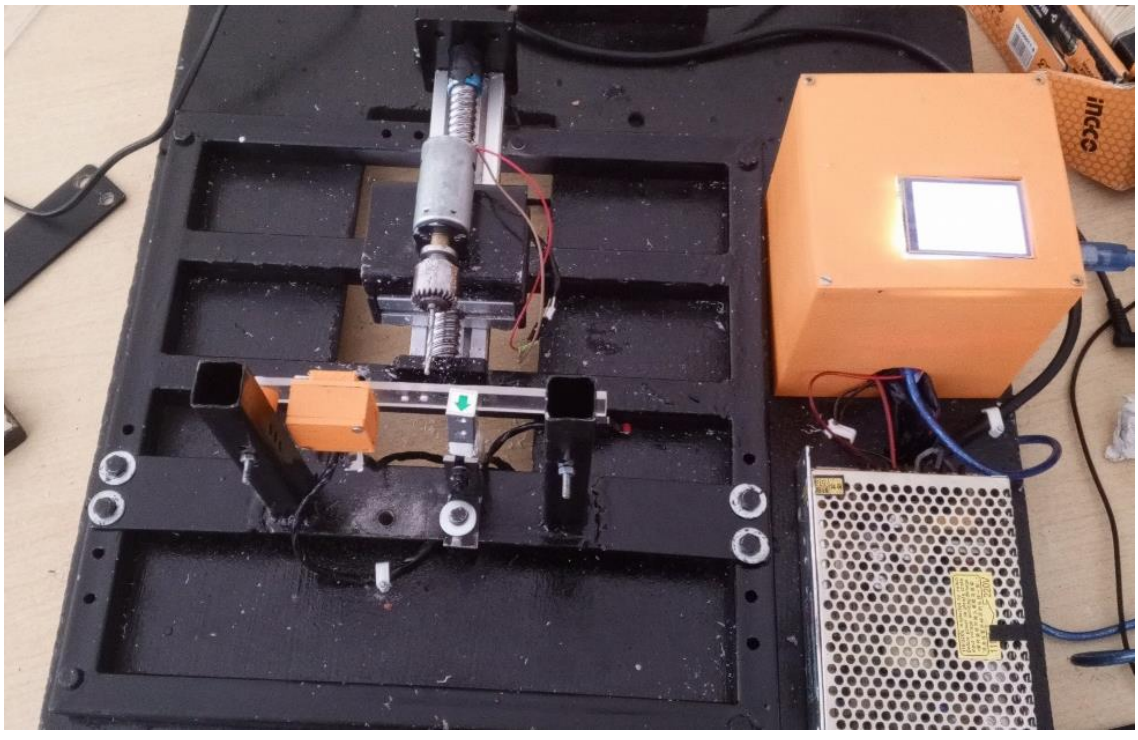


Figure 14: The full project assembly:

Figure 15: The full project assembly side view:

3.5.1 Project budget and PCB pinout

ITEM	ADDRESS	UNITS	COST	TOTAL COST
Nema 17 Stepper motor	Nema 17 Stepper motor, 12V 0.4A, 1.8 degree, 2 phase 6 wires | ATO.com	1	6585/=	6585/=
MLX90614ESF Non-contact Infrared Temperature Sensor Module	Amazon.com : MLX90614	1	1779/=	1779/=
801S Vibration Sensor Module	801S Vibration Sensor Module - ProtoSupplies	2	450/=	900
HX711 LOAD CELL AMPLIFIER+5Kg	HX711 LOAD CELL AMPLIFIER+5Kg – Besomi Electronics (besomi-egypt.com)	1	2,310.27	2310.27
TB6600 DC 4A 9-42V Stepper Motor Driver	Generic TB6600 DC 4A 9-42V Stepper Motor Driver CNC @ Best Price Online | Jumia Kenya	1	**5,395**	5,395
L298N MOTOR DRIVER	Shop & Buy Online | Jumia Kenya	1	**1897**	1897
12V DC MOTOR with drill chuck	Shop & Buy Online | Jumia Kenya	1	**300**	2200
12v dc power supply	Shop & Buy Online | Jumia Kenya		**3800**	4000

total				**25066**

Table 1: PCB pinout

	Signal pin	Power pin
Stepper	B7(pul),A8(dir)	N/a
L298N dc motor control	B12(EN),B13(IN1),B14(IN2)	N/A
Vibration sensor 1	A5	Pwr + gnd
Vibration sensor 2	10	Pwr + gnd
Limit switch	A11	Pwr + gnd
Encoder	B0, B1	Pwr + gnd
Temperature sensor	B8, B9	
Load cell	B4, B5	
start	A3	PWR
Stop	A1	
Bluetooth module	B6, B7, B3	Pwr + gnd
Start LED	A15	
Stop LED		

VIB+TEMP+LOAD = PWR+GND

3.5.2 Wiring block Diagram for the project on Simscape

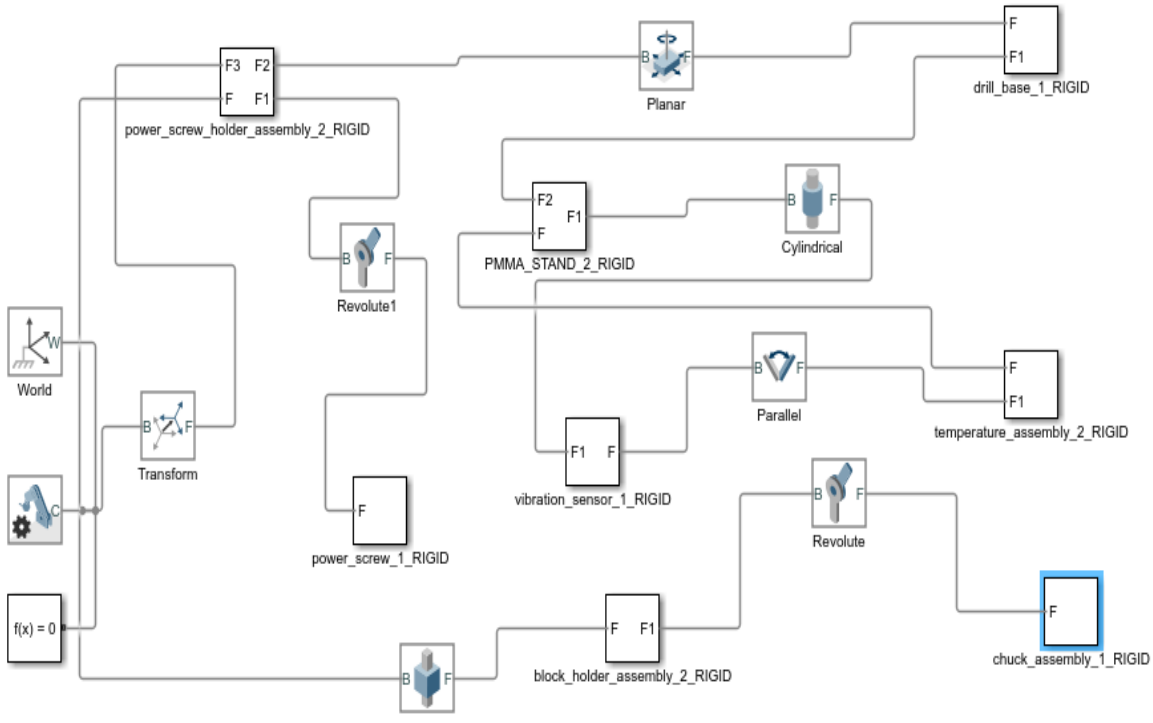


Figure 16: Wiring block Diagram for the project on Simscape

3.5.3 ELECTRICAL DATASHEET

a) L298N motor driver

The module L298n Dual H-bridge, it's often used with Arduino, it can control 2 DC motors at the same time, and you can control the direction and the speed as well. This module can control a Stepper motor as well.

This dual bidirectional motor driver, is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow you to easily and independently control two motors of up to 2A each in both directions. It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc. This board equipped with power LED indicators, on-board +5V regulator and protection diodes.

Product specification

- Input Voltage: 3.2V~40Vdc.
- Driver: L298N Dual H Bridge DC Motor Driver
- Power Supply: DC 5 V - 35 V
- Peak current: 2 Amp

- Operating current range: 0 ~ 36mA
- Operating current range: 0 ~ 36mA

  Low: -0.3V ≤ Vin ≤ 1.5V.

  High: 2.3V ≤ Vin ≤ Vss.
- Enable signal input voltage range :

  Low: -0.3 ≤ Vin ≤ 1.5V (control signal is invalid).

  High: 2.3V ≤ Vin ≤ Vss (control signal active).
- Maximum power consumption: 20W (when the temperature T = 75 ℃).
- Storage temperature: -25 ℃ ~ +130 ℃.
- On-board +5V regulated Output supply (supply to controller board i.e. Arduino).
- Size: 3.4cm x 4.3cm x 2.7cm

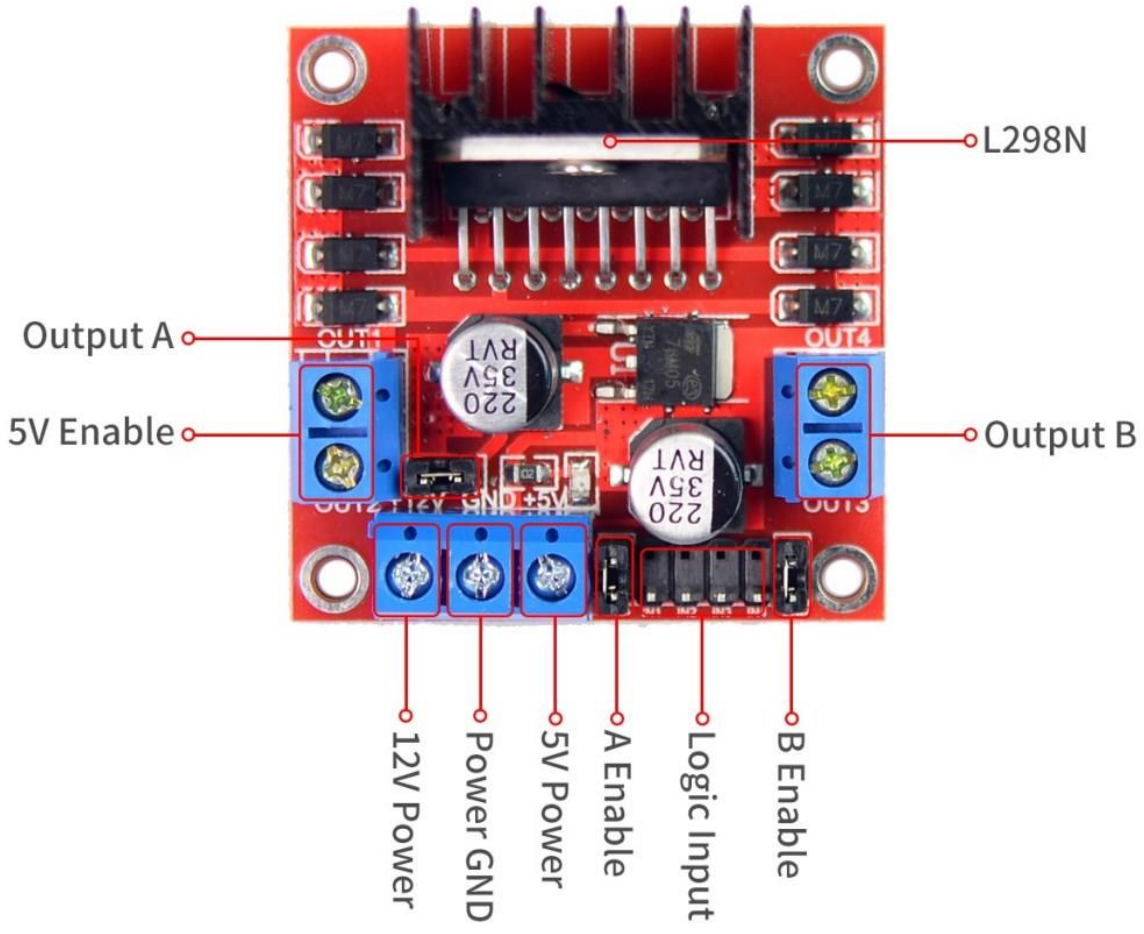


Figure 17: L298n motor driver

- The “Logic Input” pins control the directions: Forward, Backward and Stop, each two of them control A motor.

- 12V power is not always 12V it can be 9V, or it can be powered using up to 47VDC but you have to remove the regulator jumper or you'll burn it, the regulator can support only up to 12V.
- Enable A/B are for controlling the speed, if their jumpers are kept, the speed will be the maximum, they can handle up to 5V
- The module can be powered using Arduino but that's not recommended at all, it's better to use external power, and you can power the Arduino through the module too via the 5V/Gnd pins
- GND pin should always be wired with Arduino.

b) Load cell sensor

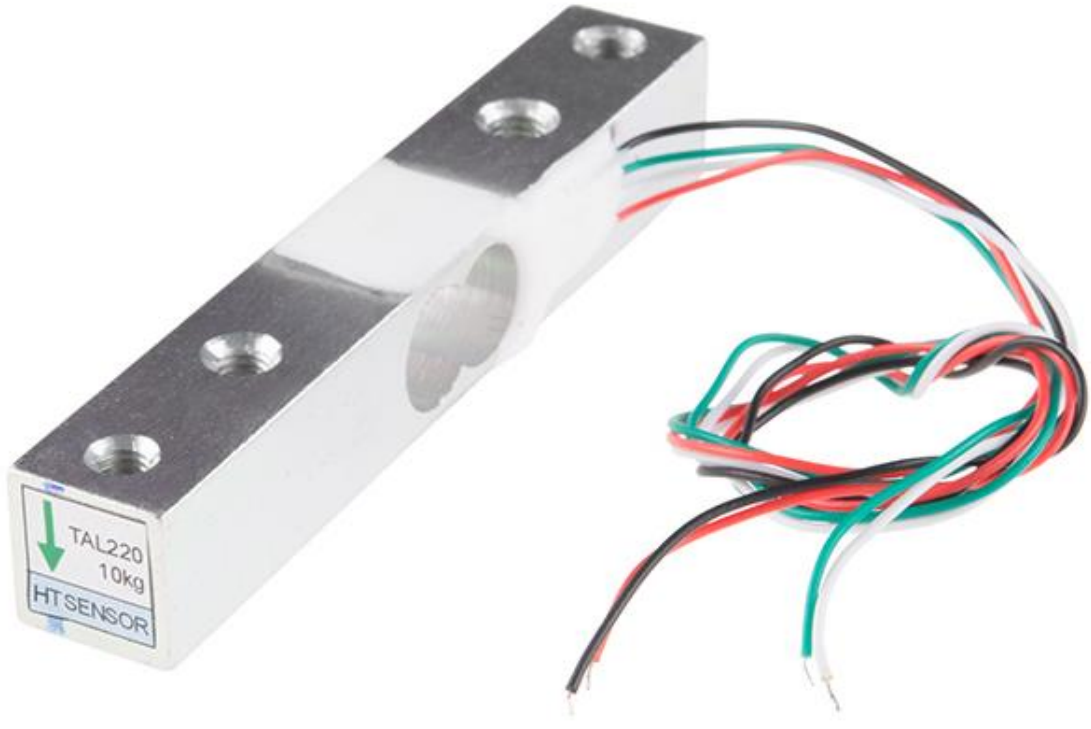


Figure 18: *Load cell*

A load cell is a force sensing module carefully designed metal structure, with small elements called strain gauges mounted in precise locations on the structure. Load cells are designed to measure a specific force,

and ignore other forces being applied. The electrical signal output by the load cell is very small and requires specialized amplification.

Installation

It's mounted by bolting down the end of the load cell where the wires are attached, and applying force on the other end in the direction of the arrow. Where the force is applied is not critical, as this load cell measures a shearing effect on the beam, not the bending of the beam. If you mount a small platform on the load cell, as would be done in a small scale, this load cell provides accurate readings regardless of the position of the load on the platform.

Calibration

A simple formula is usually used to convert the measured mv/V output from the load cell to the measured force:

Measured Force = A * Measured mV/V + B (offset)

It's important to decide what unit your measured force is - grams, kilograms, pounds, etc. This load cell has a rated output of 1.0±0.15mv/v which corresponds to the sensor's capacity of 20kg.

To find A we use

Capacity = A * Rated Output

A = Capacity / Rated Output

A = 20 / 1.0

A = 20

Since the Offset is quite variable between individual load cells, it's necessary to calculate the offset for each sensor. Measure the output of the load cell with no force on it and note the mv/V output measured by the PhidgetBridge.

Offset = 0 - 20 * Measured Output

Product specification

Mechanical	
Housing Material	Aluminum Alloy
Load Cell Type	Strain Gauge
Capacity	20kg
Dimensions	55.25x12.7x12.7mm
Mounting Holes	M5 (Screw Size)
Cable Length	550mm
Cable Size	30 AWG (0.2mm)
Cable - no. of leads	4
Electrical	
Precision	0.05%
Rated Output	1.0±0.15 mv/V
Non-Linearity	0.05% FS
Hysteresis	0.05% FS
Non-Repeatability	0.05% FS
Creep (per 30 minutes)	0.1% FS
Temperature Effect on Zero (per 10°C)	0.05% FS
Temperature Effect on Span (per 10°C)	0.05% FS
Zero Balance	±1.5% FS
Input Impedance	1130±10 Ohm
Output Impedance	1000±10 Ohm
Insulation Resistance (Under 50VDC)	≥5000 MOhm
Excitation Voltage	5 VDC
Compensated Temperature Range	-10 to ~+40°C
Operating Temperature Range	-20 to ~+55°C
Safe Overload	120% Capacity
Ultimate Overload	150% Capacity

Figure 19:Datasheet load cell

c) HX711 Load cell amplifier

The HX711 Load Cell Amplifier accepts five wires from the load cell. These pins are labeled with colors; **RED, BLK, WHT, GRN, and YLW**.

These colors correspond to the conventional color coding of load cells, where red, black, green and white wires come from the strain gauge on the load cell and yellow is an optional ground wire that is not hooked up to the strain gauge but is there to ground any small outside EMI (electromagnetic

interference). Sometimes instead of a yellow wire there is a larger black wire, foil, or loose wires to shield the signal wires to lessen EMI.

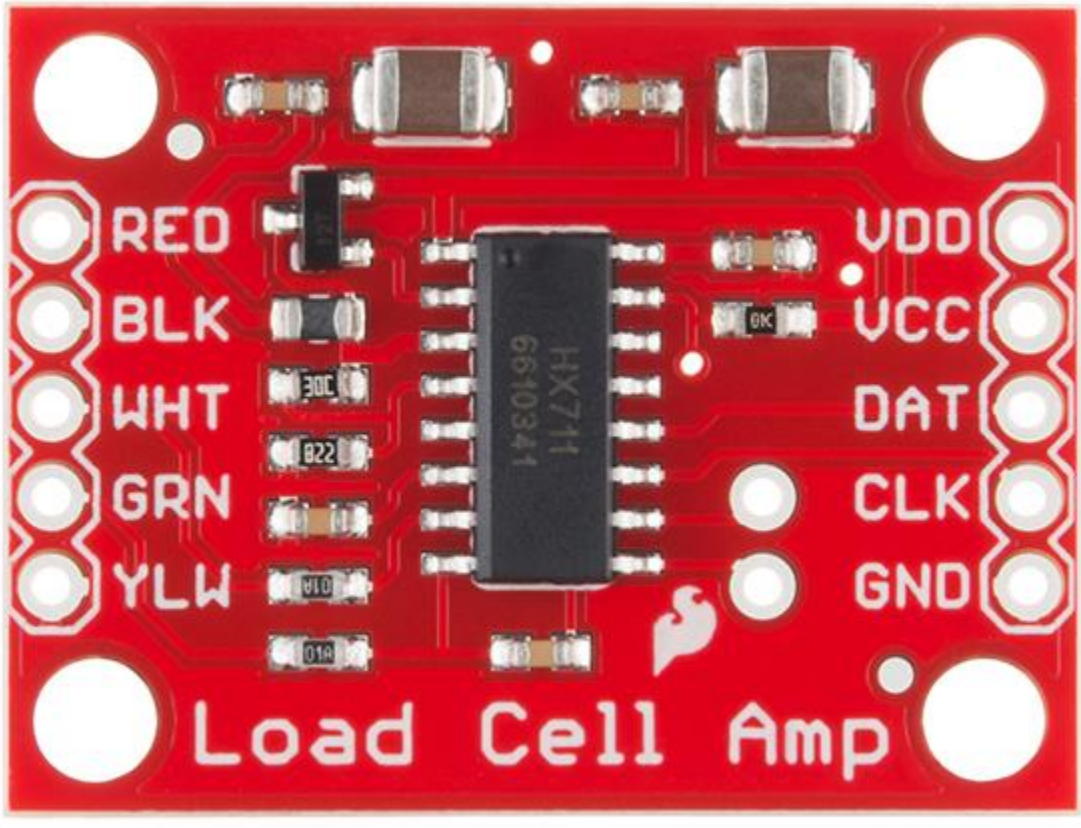


Figure 20: *Load cell amplifier*

In general, each load cell has four strain gauges that are hooked up in a wheatstone bridge formation as shown below.

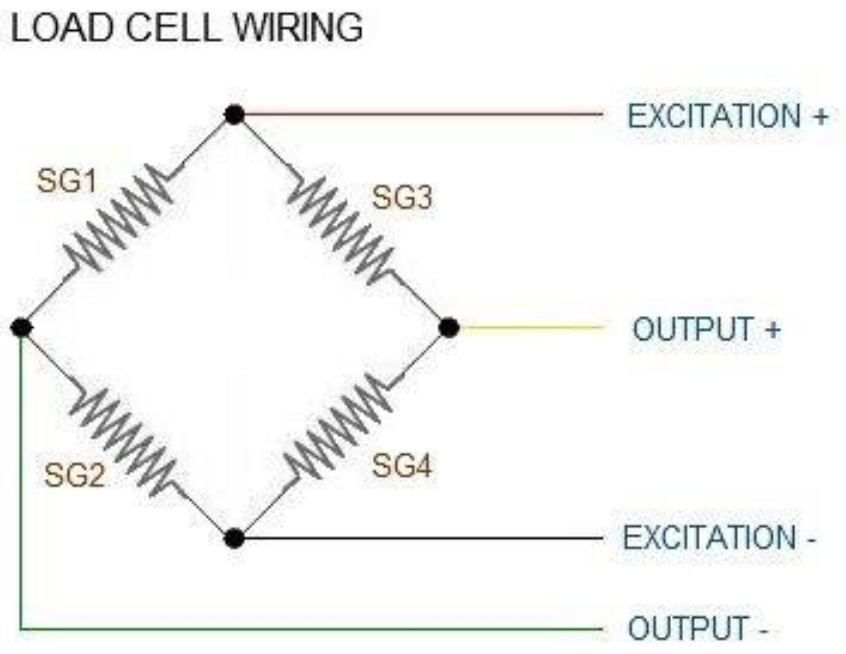


Figure 21: Load cell wiring

Table 2: Wheatson bridge

Wheatstone Bridge Node	**"Typical" Wire Color**
Excitation+ (E+) or VCC	RED
Excitation- (E-) or GND	BLACK or YELLOW
Output- (O-), Signal- (S-), or Amplifier- (A-)	WHITE
O+, S+, or A+	GREEN or BLUE

Some load cells might have slight variations in color coding such as blue instead of green or yellow instead of black or white if there are only four wires (meaning no wire used as an EMI buffer). You might have to infer a little from the colors that you have or check the datasheet on the load cell, but in general you will usually see these colors.

Once the load cell is hooked up to the amplifier, you can hook up VDD, VCC, DAT, CLK, and GND to a microcontroller such as Arduino board.

Note: VCC is the analog voltage to power the load cell. VDD is the digital supply voltage used to set the logic level.

Serial Interface

Pin PD_SCK and DOUT are used for data retrieval, input selection, gain selection and power down controls. When output data is not ready for retrieval, digital output pin DOUT is high. Serial clock input PD_SCK should be low. When DOUT goes to low, it indicates data is ready for retrieval. By applying 25~27 positive clock pulses at the PD_SCK pin, data is shifted out from the DOUT output pin. Each PD_SCK pulse shifts out one bit, starting with the MSB bit first, until all 24 bits are shifted out. The 25th pulse at PD_SCK input will pull DOUT pin back to high (Figure 1). Input and gain selection are controlled by the number of the input PD_SCK pulses (Table 1). PD_SCK clock pulses should not be less than 25 or more than 27 within one conversion period, to avoid causing serial communication error.

PD_SCK Pulses	Input channel	Gain
25	A	128
26	B	32
27	A	64

Figure 22: Pinouts

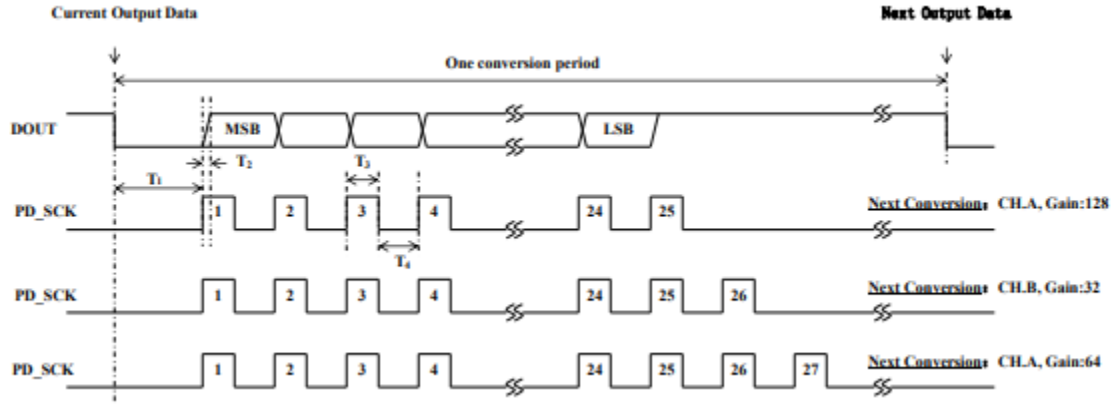


Figure 23: Data output, input and gain selection timing and control

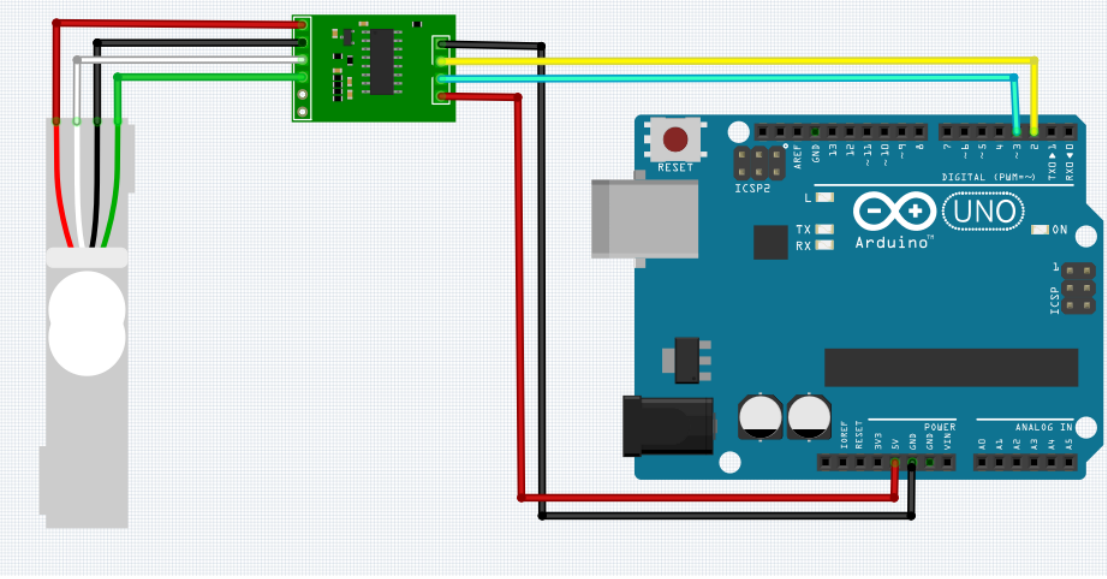


Figure 24: Load cell

d) Vibration sensor

This module features an adjustable potentiometer, a vibration sensor, and a LM393 comparator chip to give an adjustable digital output based on the amount of vibration.

The potentiometer can be adjusted to both increase and decrease the sensitivity to the desired amount. The module outputs a logic level high (VCC) when it is triggered and a low (GND) when it isn't. Additionally, there is an onboard LED that turns on when the module is triggered.

Features

- The default state of the swith is close
- Digital output Supply voltage:3.3V-5V
- On-board indicator LED to show the results
- On-board LM393 chip
- SW-420 based sensor, normally closed type vibration sensor
- Dimension of the board: 3.2cm x 1.4cm

This sensor module produce logic states that depends on vibration and external force applied on it. When there is no vibration this module gives logic LOW output. When it feels vibration then output of this

module goes to logic HIGH. The working bias of this circuit is between 3.3V to 5V DC.

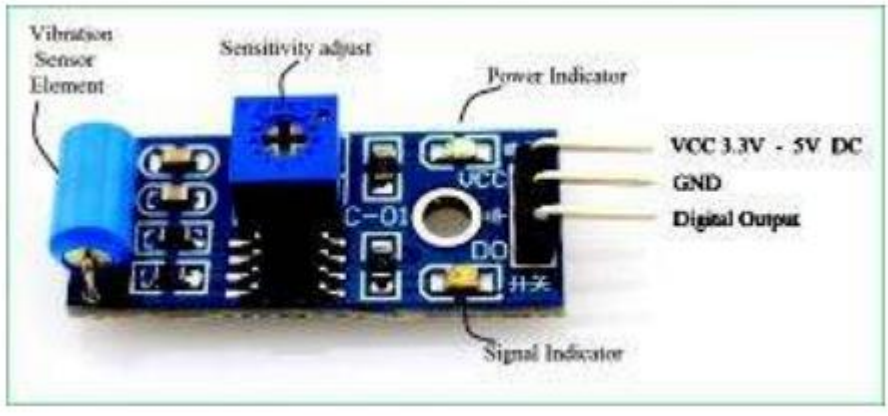


Figure 25: typical vibration sensor

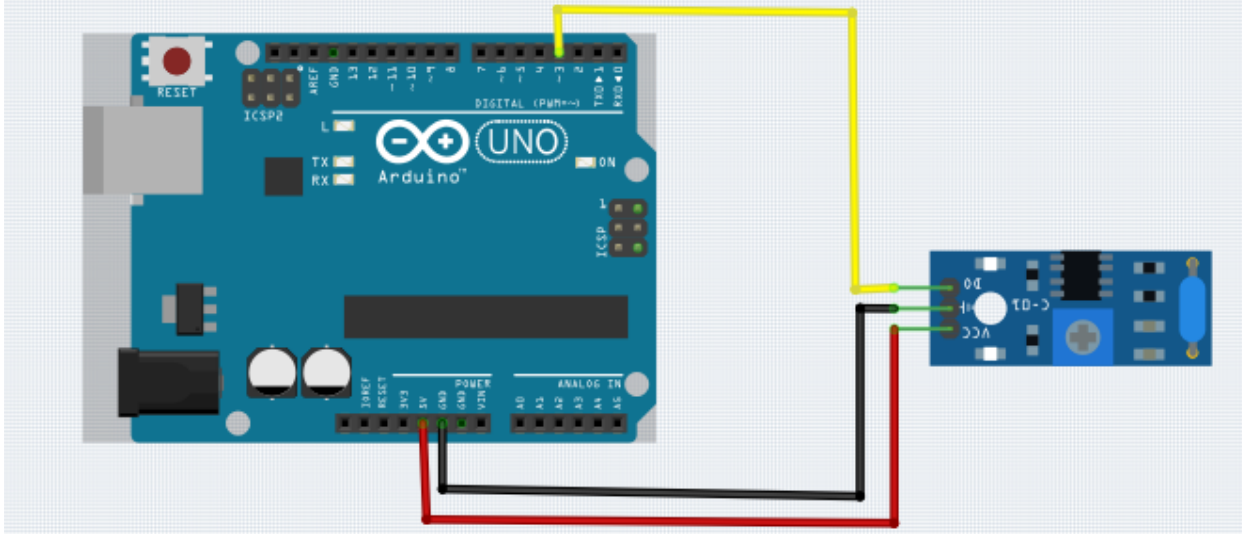


Figure 26: vibration sensor wiring

Sensor Details

SW-420 Single-roller type full induction trigger switch. When no vibration or tilt, the product is ON conduction state, and in the steady state, when a vibration or tilt, the switch will be rendered instantly disconnect the conductive resistance increases, generating a current pulse signal, thereby triggering circuit. These products are completely sealed package, waterproof, dustproof.

Principle

Usually at any angle switch is ON state, by the vibration or movement, the rollers of the conduction current in the switch will produce a movement or vibration, causing the current through the disconnect or the rise of the resistance and trigger circuit. The characteristics of this switch is usually general in the

conduction state briefly disconnected resistant to vibration, so it's high sensitivity settings by IC, customers according to their sensitivity requirements for adjustments.

e) MLX90614 Temperature sensor

MLX90614 is IR based contactless temperature sensor that can measure the temperature of a particular object between -70°C – 382.2°C and an ambient temperature of -40°C – 125°C without even making physical contact with an object under observation. It is embedded with an I2C port to communicate temperature reading to microcontrollers over an I2C bus. On top of that, it is provided with ESD protection to avoid malfunctioning of the sensor.

The tiny device is highly accurate and precise due to its powerful ADC. A 17-bit ADC is embedded in the module to output the values with 0.14 ˚C of resolution. Melexis has introduced different versions of this sensor based on input voltage requirements i.e., 3 Volts or 5volts and resolving power for different project requirements. But MLX90614 is a sensitive temperature sensor that has a long list of applications, especially in home automation

MLX90614 Pinout

This temperature sensor module comes with a 3.3 voltage regulator, I2C Bus with internal pullup resistors to define a default state and a capacitor for noise filtering. The pinout of the non-contact MLX90614 IR Temperature Sensor module is as shown:

Pin Configuration

MLX90614 has two versions and is available in the TO-39 package. The pin configuration details are listed in a table below:

Pin Name	Function
VCC	Positive power supply pin
GND	Reference potential pin
SCL	Open drain Serial Clock pin. An I2C line clock pulses pin for data synchronization.
SDA	Open drain Serial Data pin. An I2C line to communicate data to the host MCU.

Features & Specifications

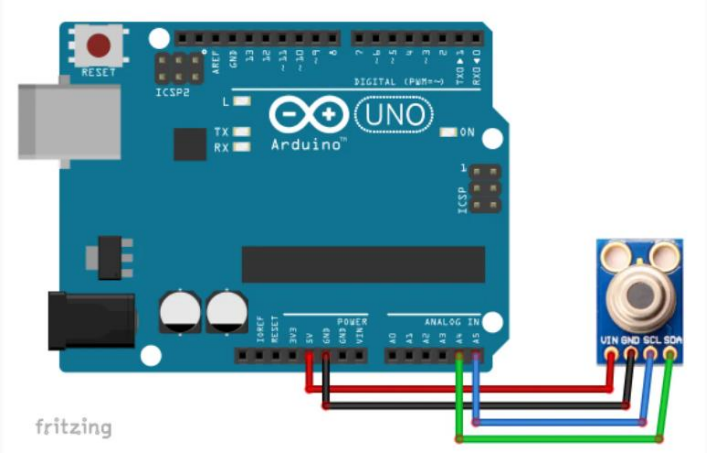
- Operating Voltage: 3.6 Volts – 5.5 Volts

- Ambient Temperature Range: -40°C – 125°C
- Object Temperature Range: -70°C – 380°C
- Measurement resolution: 0.02°C
- ESD Sensitivity: 2kV
- Sink/Source Current: 25mA
- ADC Resolution: 17 bits
- IR sensor is integrated with an optical filter, a DSP, and a low noise amplifier for fine output digital signals.
- Adaptable for 8-16 Volts applications and can be integrated easily.
- Supports power-saving mode and is available in single and dual versions
- It is a power-efficient and highly sensitive sensor.

Connection Diagram

The following figure shows the connection diagram between MLX90614 temperature sensor and Arduino.

- Connect the power supply pin (Vin) of the temperature sensor to the 5V pin of Arduino and the GND pin of MLX90614 to the GND pin Arduino UNO.
- Connect the SDA and SCL pins of the mentioned IR sensor to the A4 and A5 pins of the Arduino UNO for transferring data serially. A4 and A5 pins of Arduino also share alternate function of SDA and SCL pins of I2C port of Arduino Uno.



Arduino UNO	MLX90614 IR Sensor
5V	VCC
GND	GND
SDA	A4

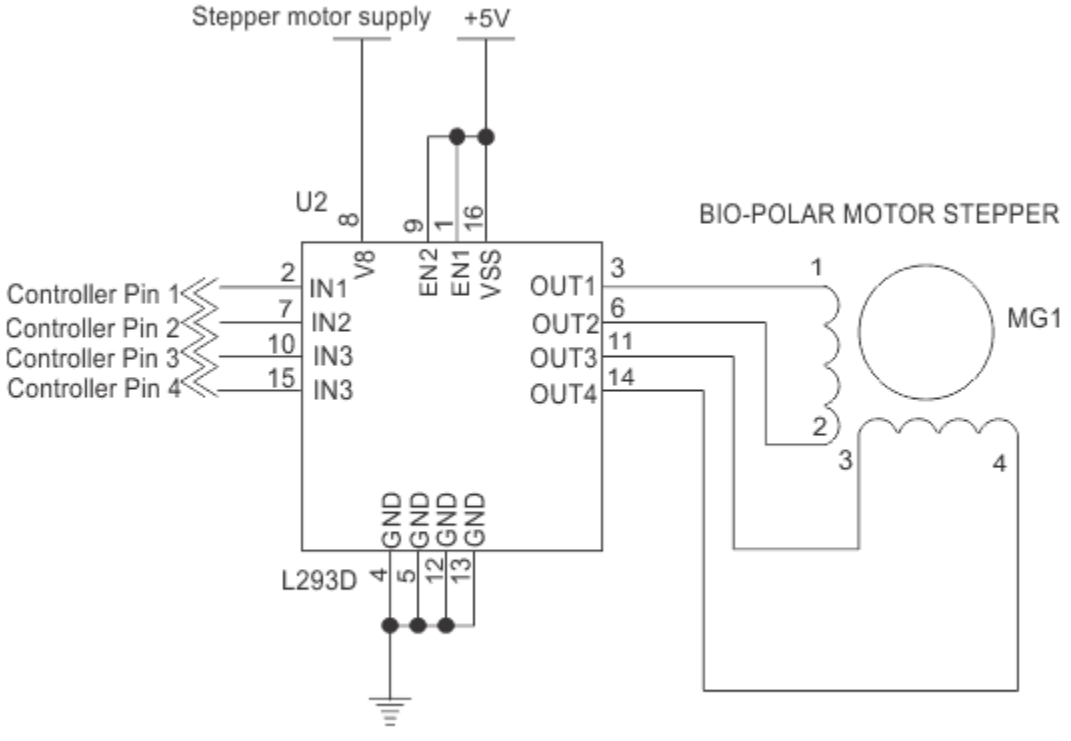
SCL	A5

f) Bipolar stepper motor

A stepper motor is a brushless DC motor that divides a full rotation into several equal steps of 1.8 degrees thus they can make up to 200 steps in one rotation. The motor's position can then be directed to move and hold at one of these steps without any position sensor for feedback.
Bipolar stepper motors are a type of stepper motor with a single winding per phase and no center tap unlike a unipolar stepper motor.
The DC current in a winding needs to be reversed to reverse a magnetic pole and allow the motor to function. A H-bridge IC is used to drive the bipolar stepper motor because of the following reasons;

- They can reverse the polarity of the stator coils.
- They are capable of handling high current that is drawn by the stepper motor since the microcontroller could only handle up to 15 mA.
- They can protect the microcontroller pins against high spikes that results when the coil current changes direction.

Bipolar motor driver circuit interface



Interfacing to H bridge IC L2093D

3.6 GUI Design for data collection, visualization and motor control

To be able to control the speed of the drilling motor and to visualize in real time the different parameters vary during the experiments, we developed a graphical user interface using Python. The GUI uses the

serial communication protocol to read data and send commands to the modules to control motor action. The capabilities of the graphical user interface included:

- Selection of the COM ports to enable communication with the microcontrollers.
- Switching between light and dark mode to provide comfort of use to the user.
- Control of the dc motor speed used for a drilling experiment
- Real time monitoring of the sensor outputs coupled with a graphical interface to visualize the changes of temperature, vibrations and load applied throughout the experiment.
- Generation of csv files to save sensor data for analysis.
- Ability to start and stop motors and data collection at any time.

a) Read Sensors Graphical User Interface

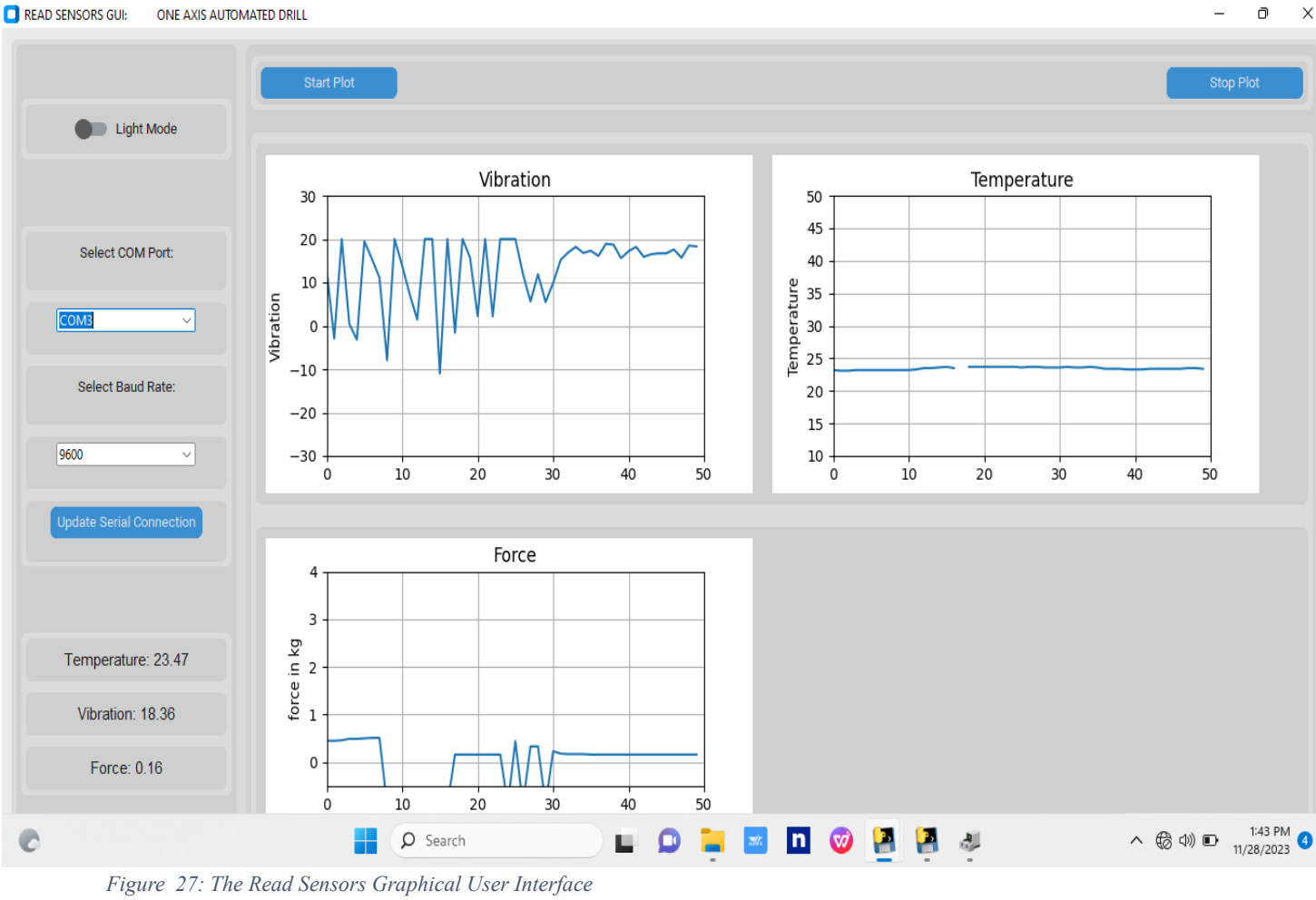


Figure 27: The Read Sensors Graphical User Interface

b) Motor Control Graphical User Interface

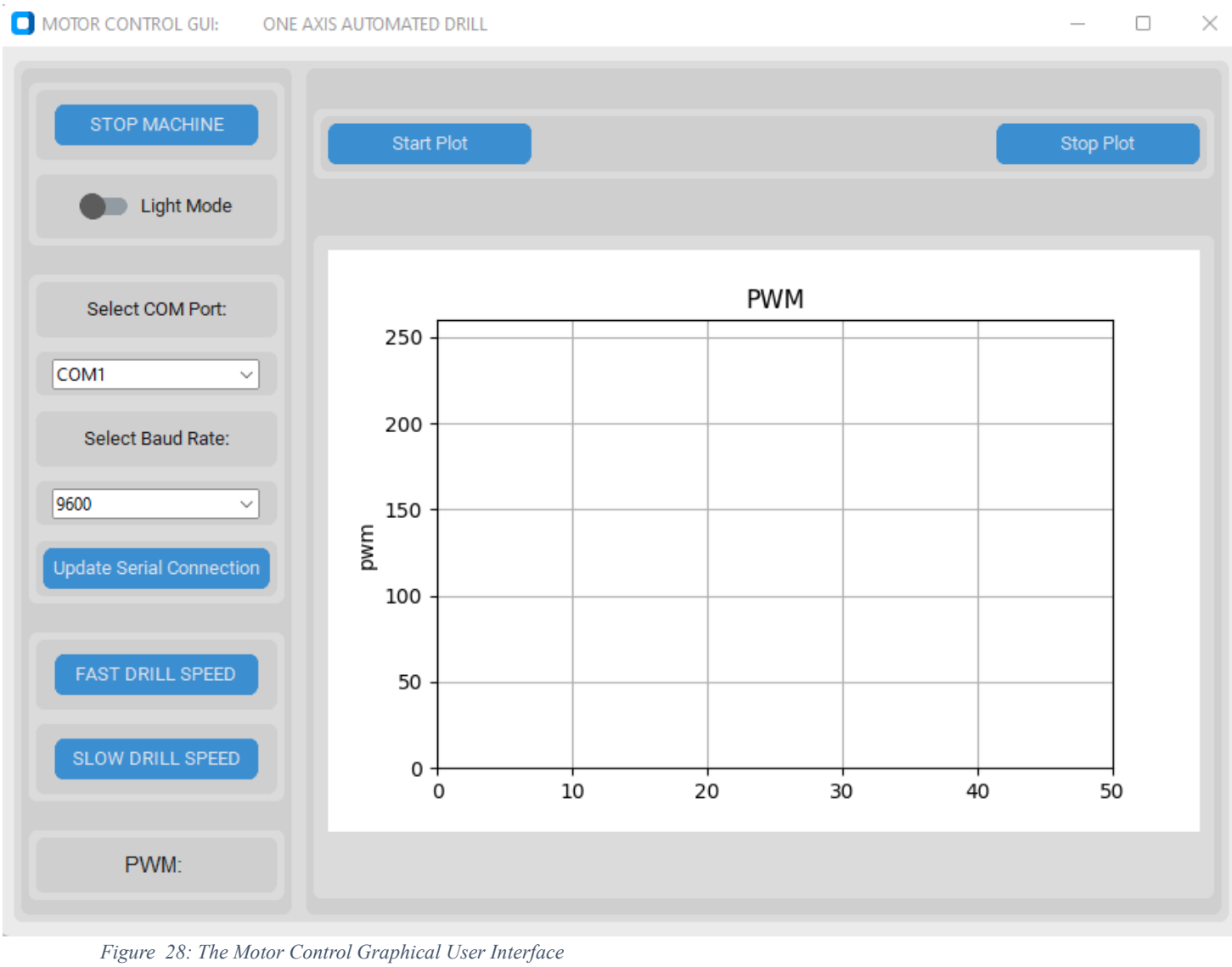


Figure 28: The Motor Control Graphical User Interface

3.7 Sample data collected:

A	B	C	D
Vibration	Temperatu	Force	
45.91	20.04	-0.01	
46.05	15.42	0.07	
46.39	15.85	-0.07	
46.47	18.67	0	
46.53	14.79	-0.01	
46.75	16.04	-0.01	
46.81	19.61	-0.01	
46.81	18.91	-0.01	
47.07	20.04	-0.01	
47.35	14.16	-0.01	
47.35	15.49	-0.02	
47.35	16.51	-0.01	
47.51	14.32	-0.01	
47.55	18.63	-0.01	
47.55	17.69	-0.01	
47.63	17.97	-0.01	
47.47	17.65	-0.01	
47.53	19.65	-0.01	
47.57	14.91	-0.01	
47.67	14.2	-0.01	
47.75	14.08	-0.01	
47.93	19.53	-0.01	
48.25	18.63	-0.01	
48.37	16.87	-0.01	

Figure 29: Sample data collected

The data collected was analyzed to determine relationships between the different variables. The codes developed throughout the project can be accessed on GitHub on: https://github.com/marymbaire/ONE-AXIS-AUTOMATED-DRILL-PROJECT

Chapter 4: Presentation of Findings, Analysis and Interpretation

4.1 Introduction

In this chapter, results from the various One Axis drill experiment will first be presented. These experiments provide information about the effect of variations to the parameters regarding the model output.

The data collected was for 18 experiments broken down like this:
Long span stand experiment:

- fast drill speed (3 iterations)
- slow drill speed (3 iterations)

Medium span stand experiment:

- fast drill speed (3 iterations)
- slow drill speed (3 iterations)

Short span stand experiment:

- fast drill speed (3 iterations)
- slow drill speed (3 iterations)

As a standard, fast drill speeds run at a PWM value of 255, whereas slow drill speeds run at a PWM value of 200.

4.2 Graphs of cumulative experiments.

1. **Long span fast drill speed action iterations**

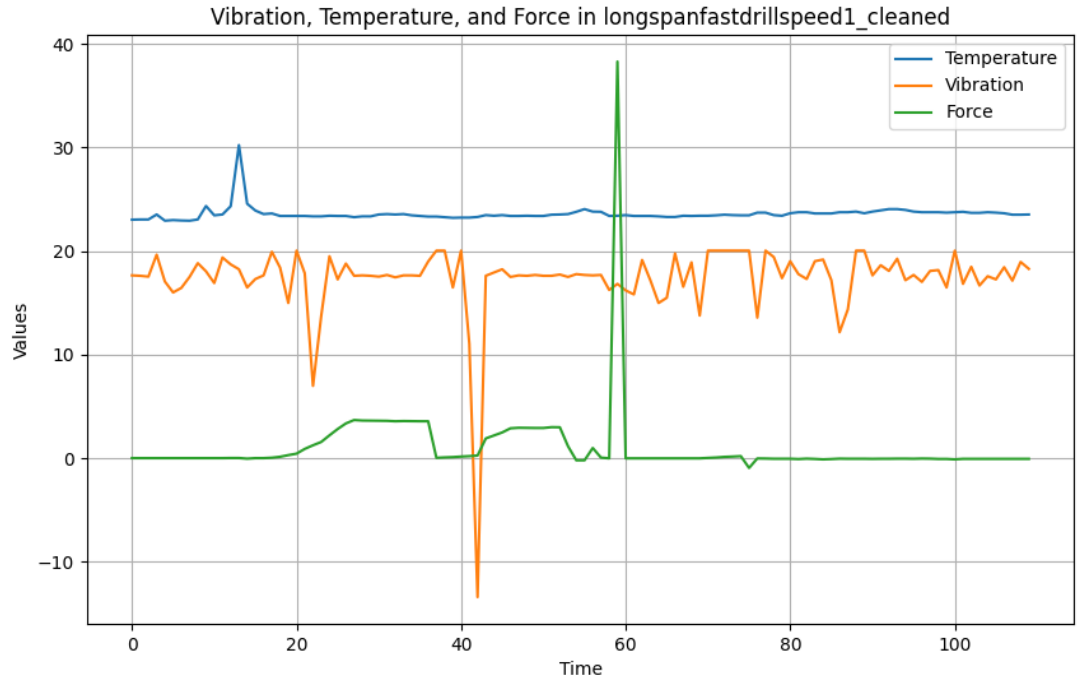


Figure 30: Vibration, Temperature, force in long span fast drill l speed iteration 1

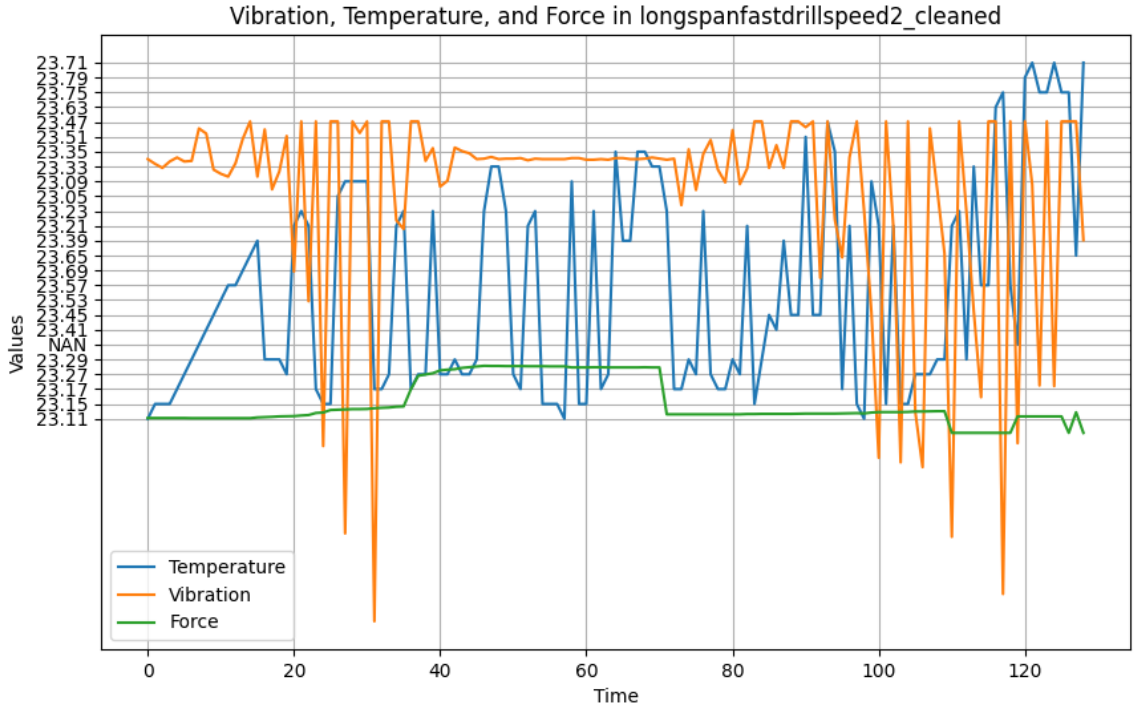


Figure 31: Vibration, Temperature, force in long span fast drill speed iteration 2

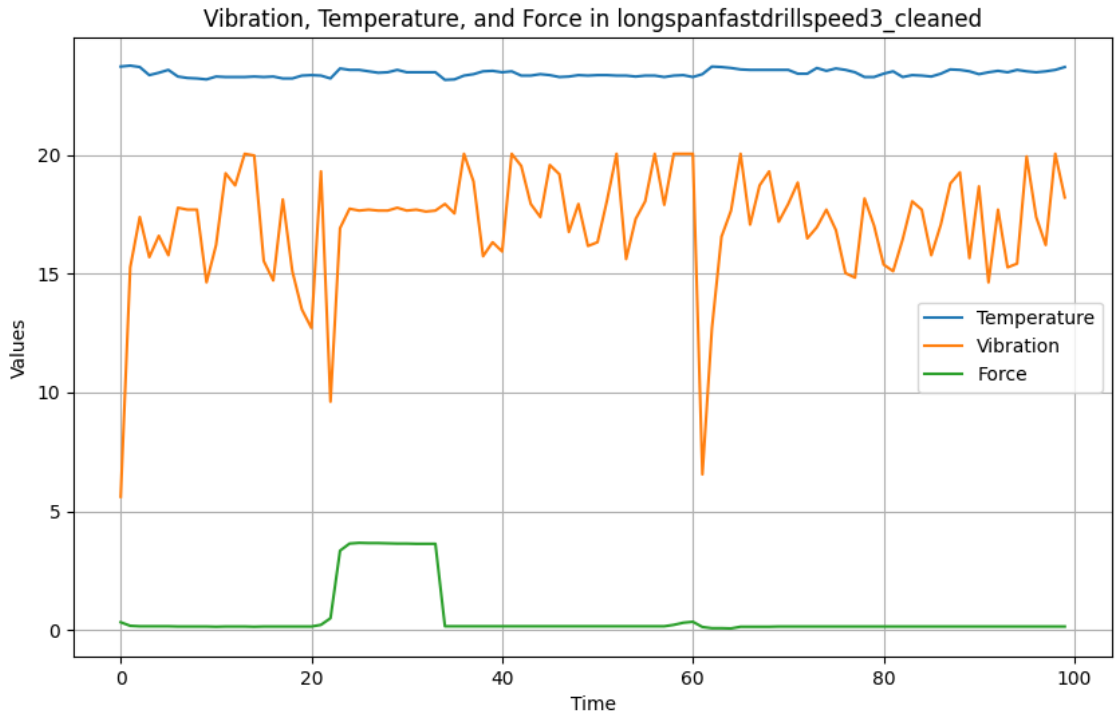


Figure 32: Vibration, Temperature, force in long span fast drill speed iteration 3

2. Long span slow drill speed action iterations

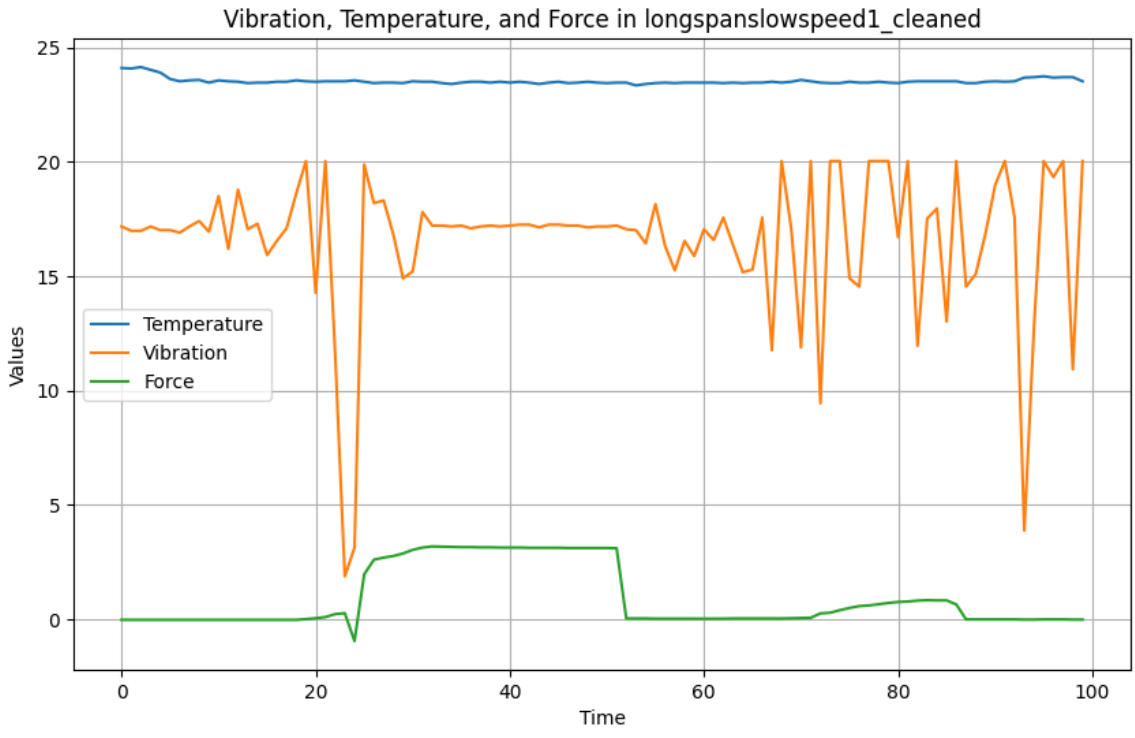


Figure 33:Vibration, Temperature, force in long span slow drill speed iteration 1

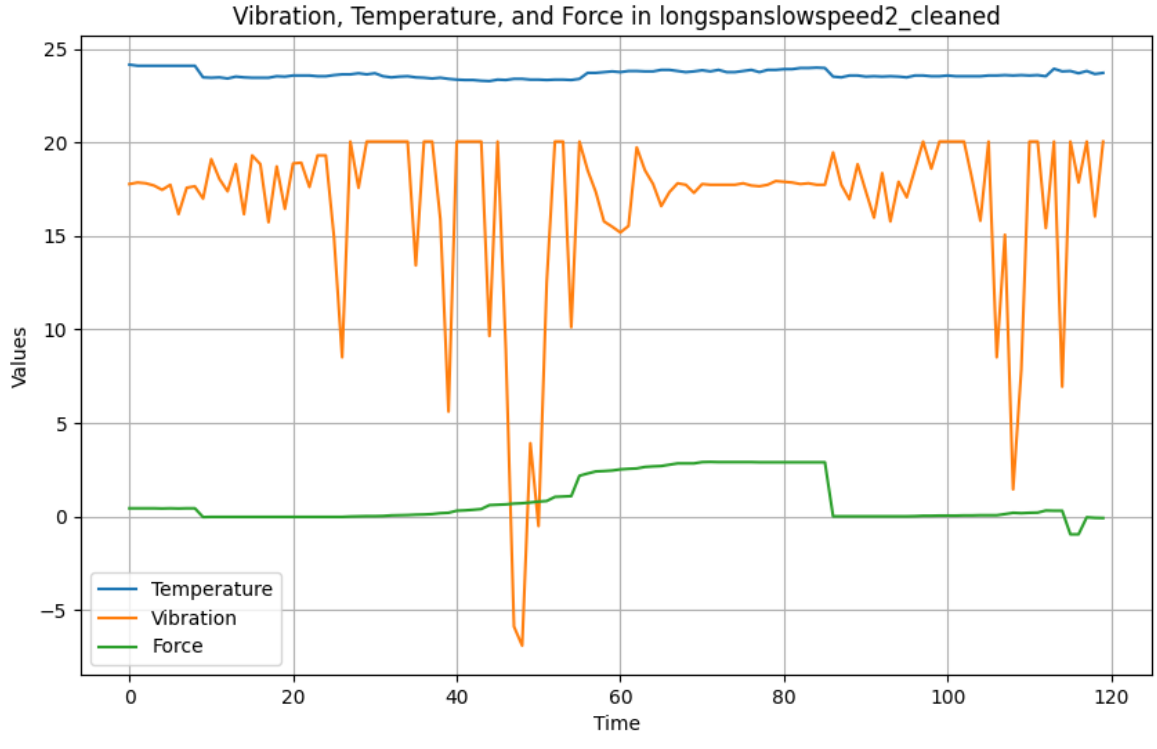


Figure 34: Vibration, Temperature, force in long span slow drill speed iteration 2

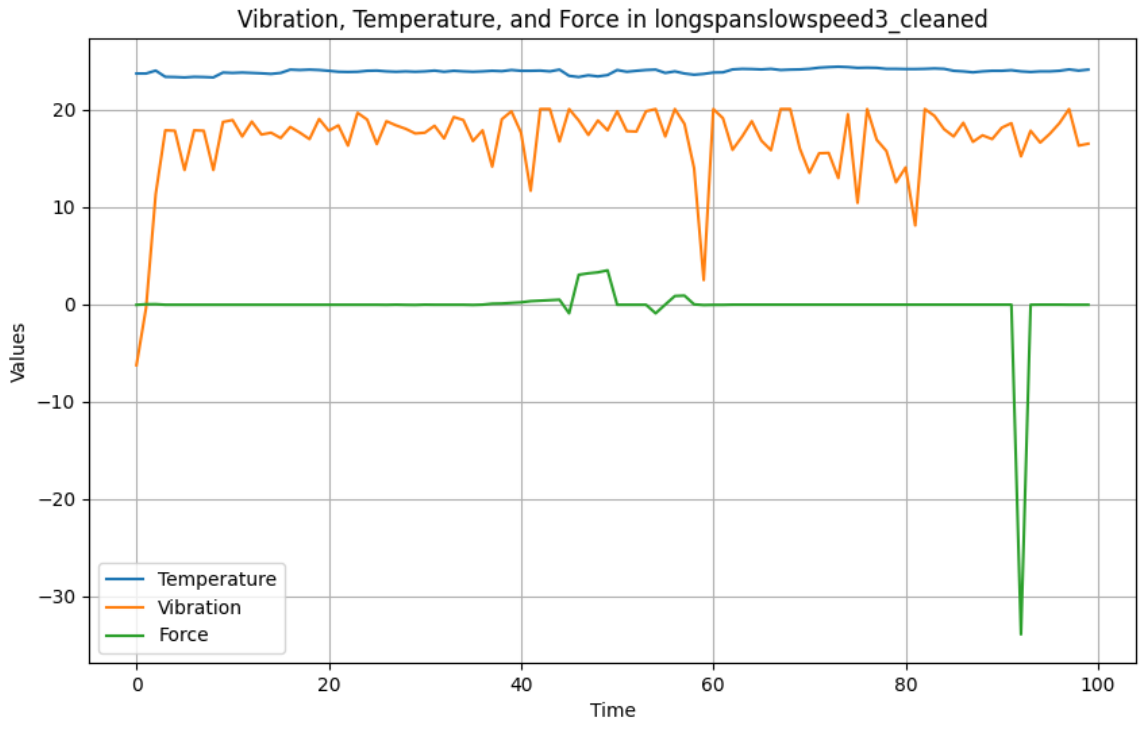


Figure 35: Vibration, Temperature, force in long span slow drill speed iteration 3

3. Medium span fast drill speed action iterations

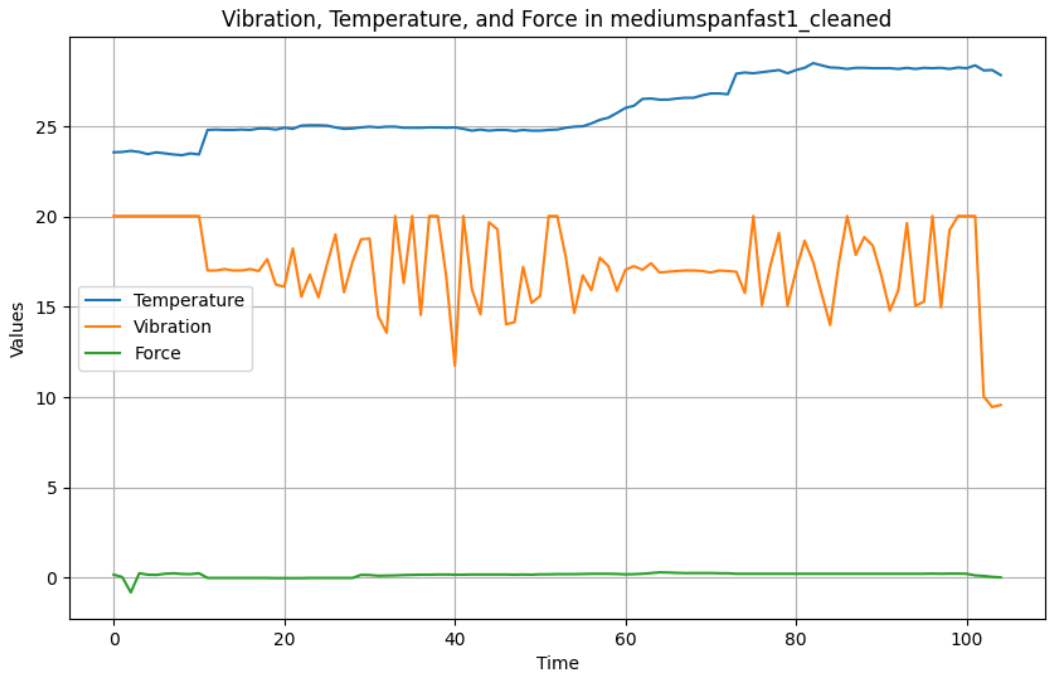


Figure 36: Vibration, Temperature, force in medium span fast drill speed iteration 1

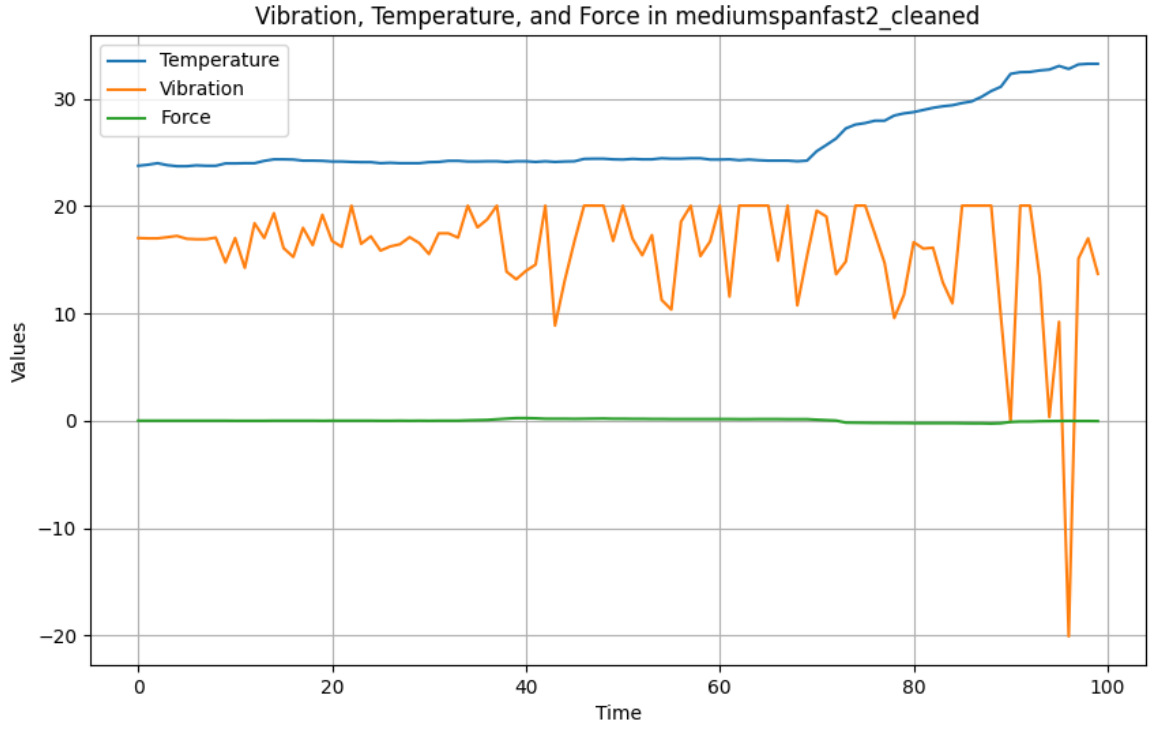


Figure 37: Vibration, Temperature, force in medium span fast drill speed iteration 2

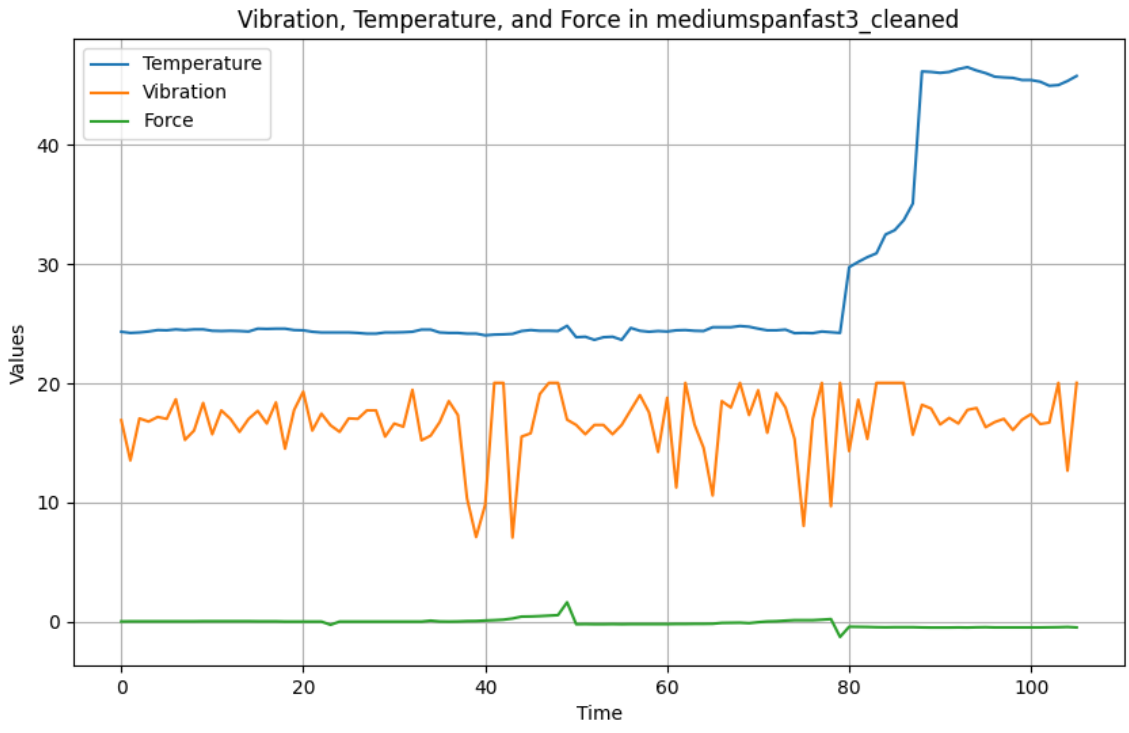


Figure 38: Vibration, Temperature, force in medium span fast drill speed iteration 3

4. Medium Span Slow Drill Speed Action Iterations

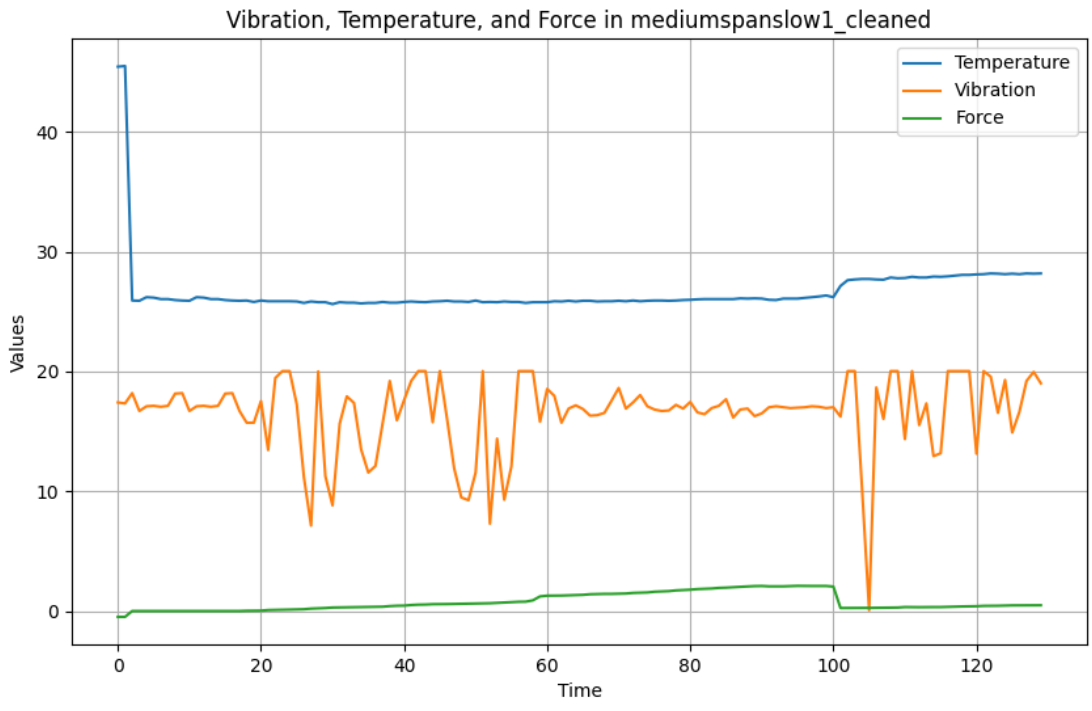


Figure 39: Vibration, Temperature, force in medium span slow drill speed iteration 1

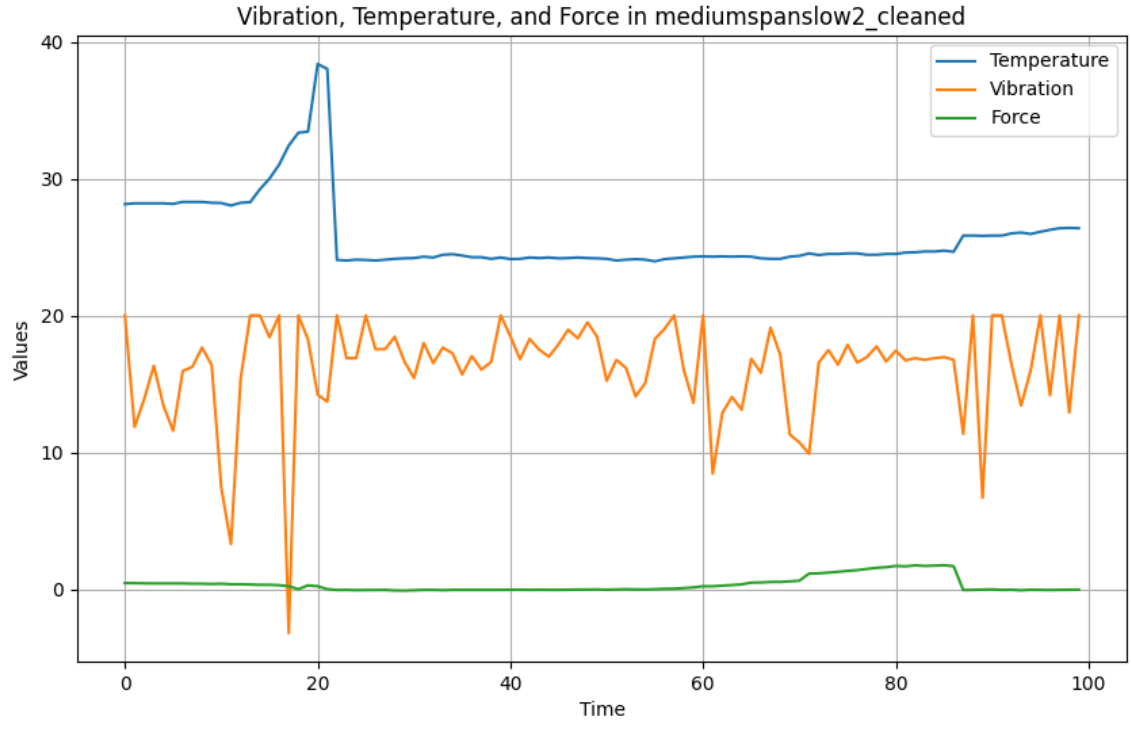


Figure 40: Vibration, Temperature, force in medium span slow drill speed iteration 2

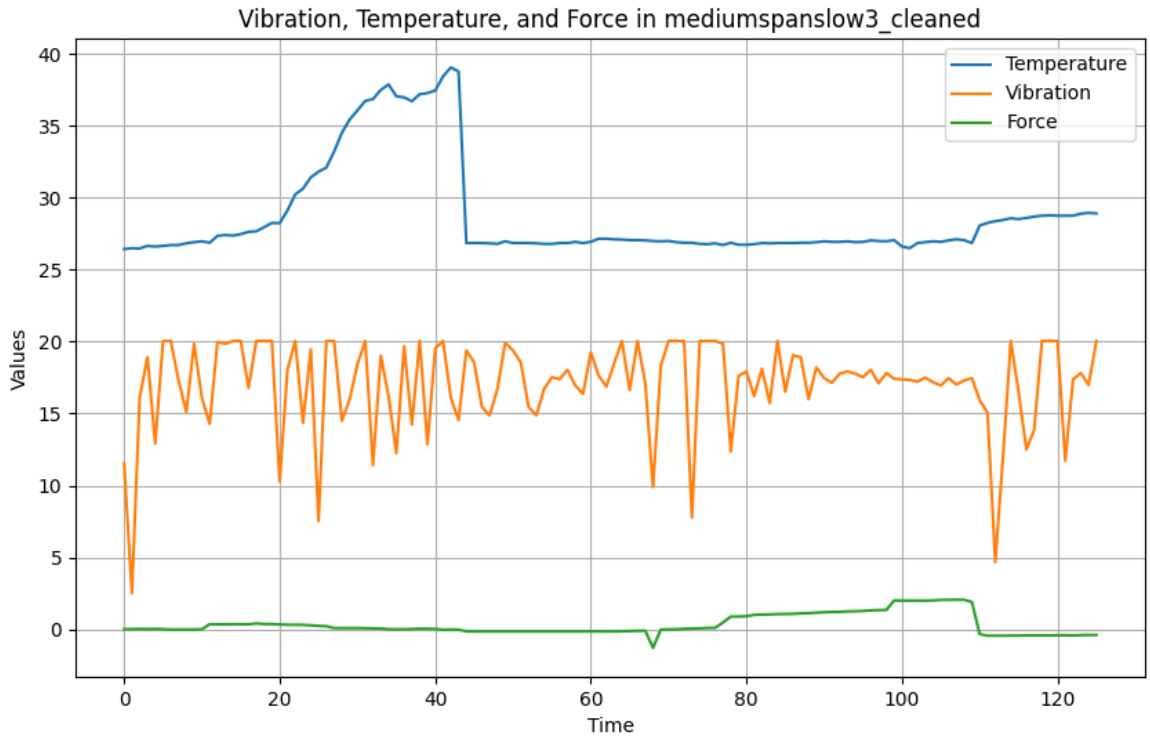


Figure 41: Vibration, Temperature, force in medium span slow drill speed iteration 3

5. Short span fast drill speed action iterations

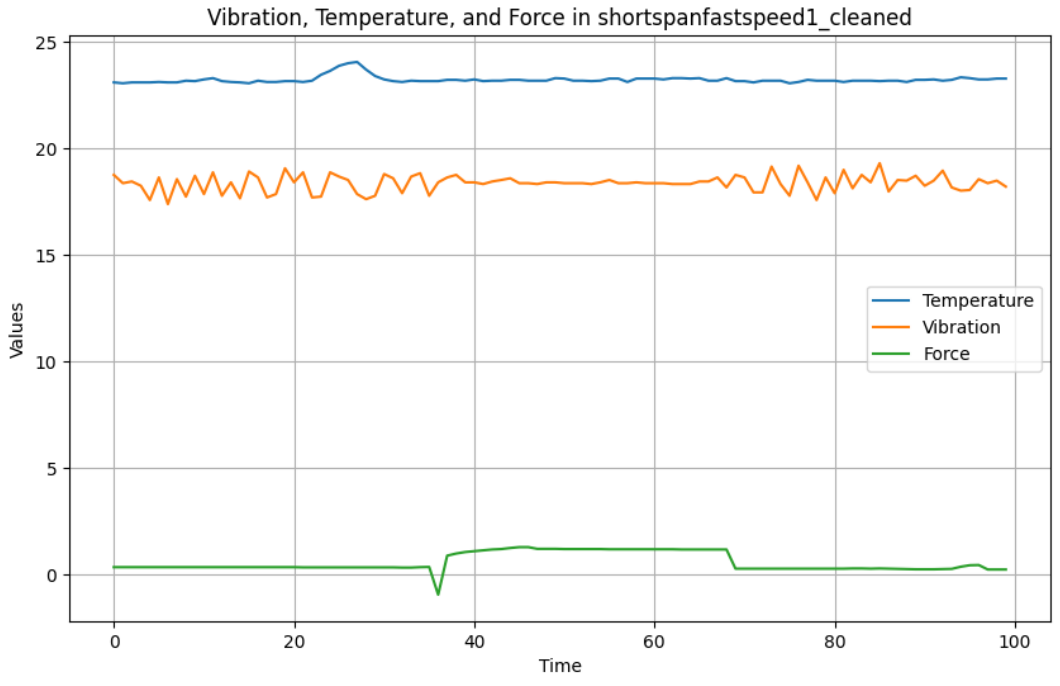


Figure 42: Vibration, Temperature, force in short span fast drill speed iteration 1

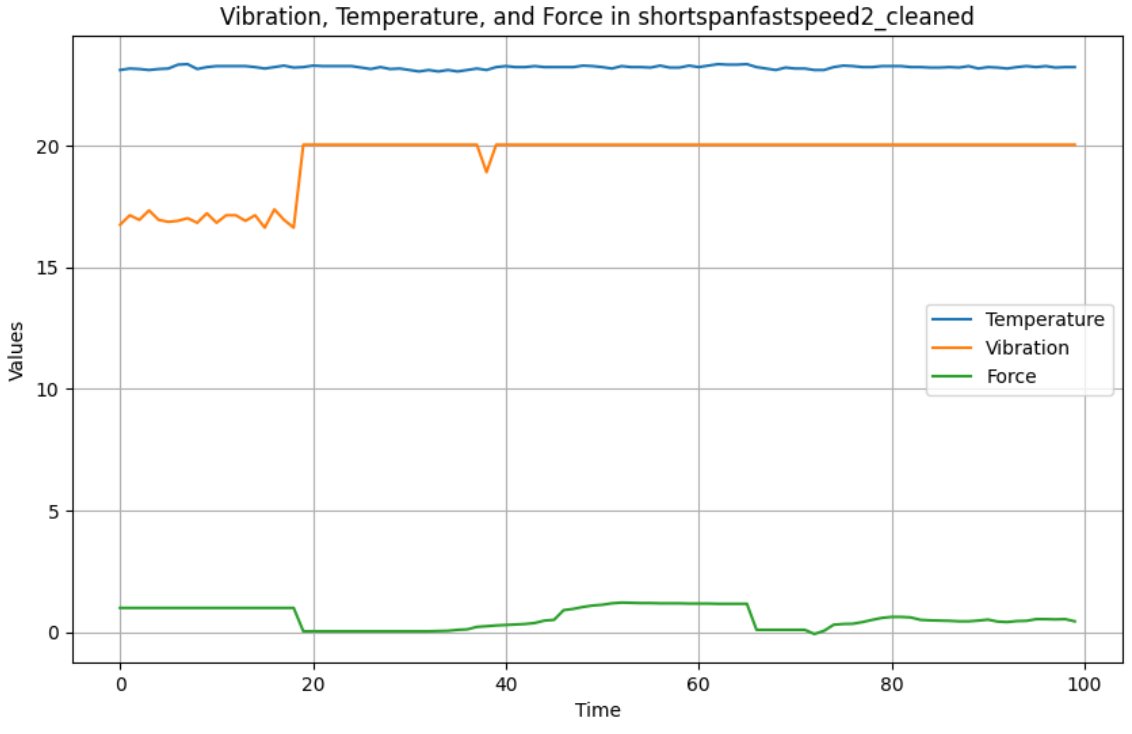


Figure 43: Vibration, Temperature, force in short span fast drill speed iteration 2

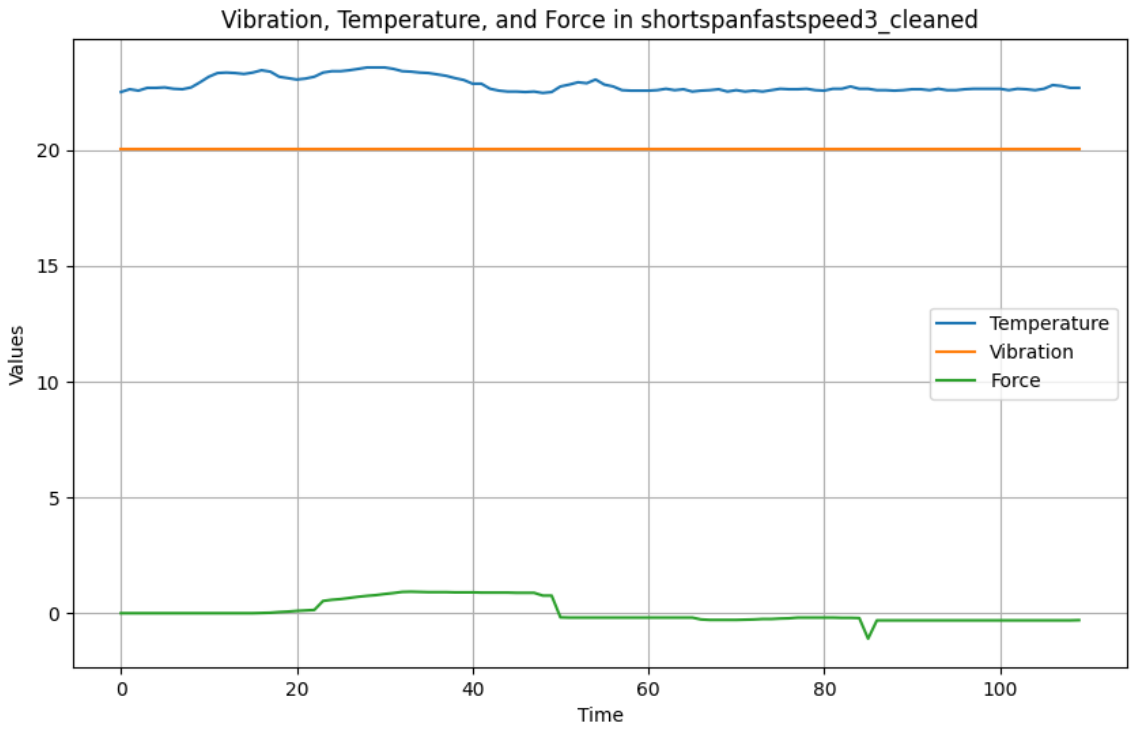


Figure 44: Vibration, Temperature, force in short span fast drill speed iteration 3

6. Short span fast drill speed action

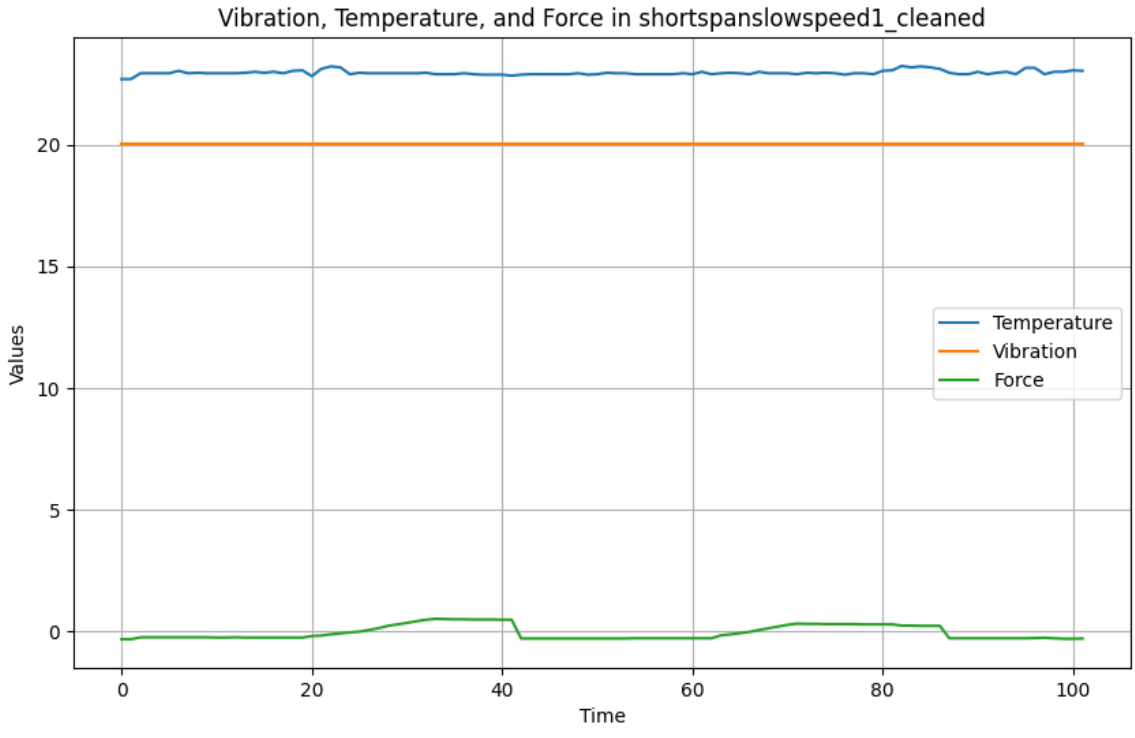


Figure 45: Vibration, Temperature, force in short span slow drill speed iteration 1

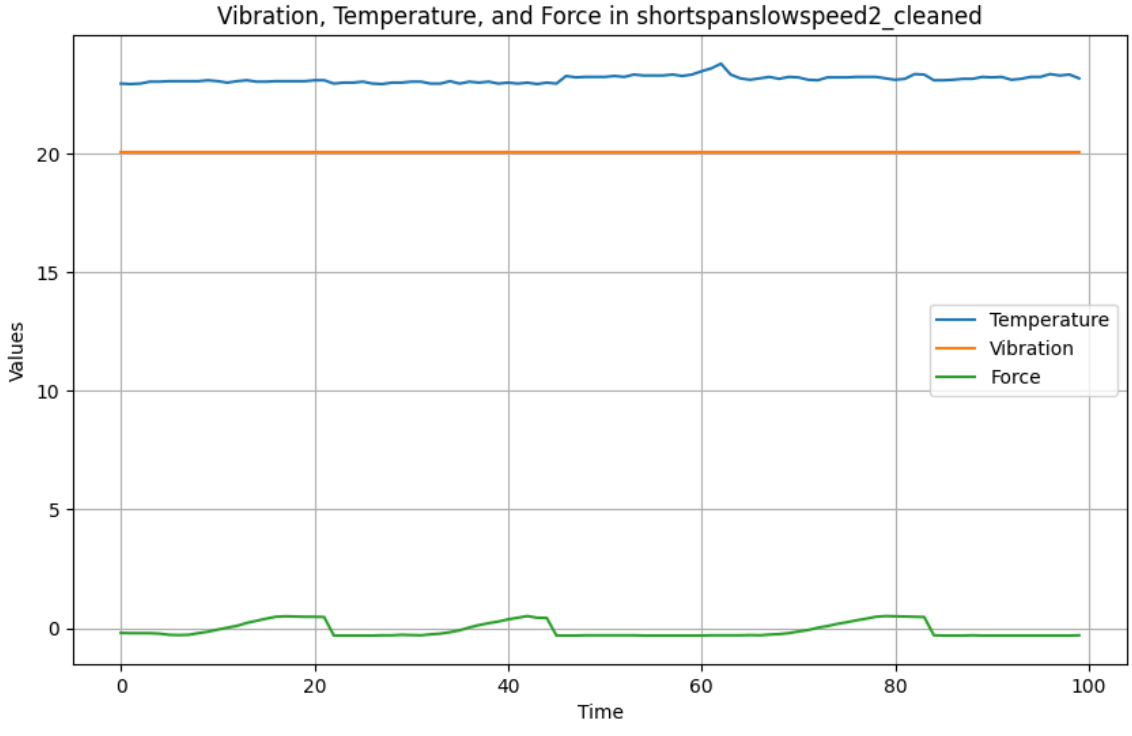


Figure 46: Vibration, Temperature, force in short span slow drill speed iteration 2

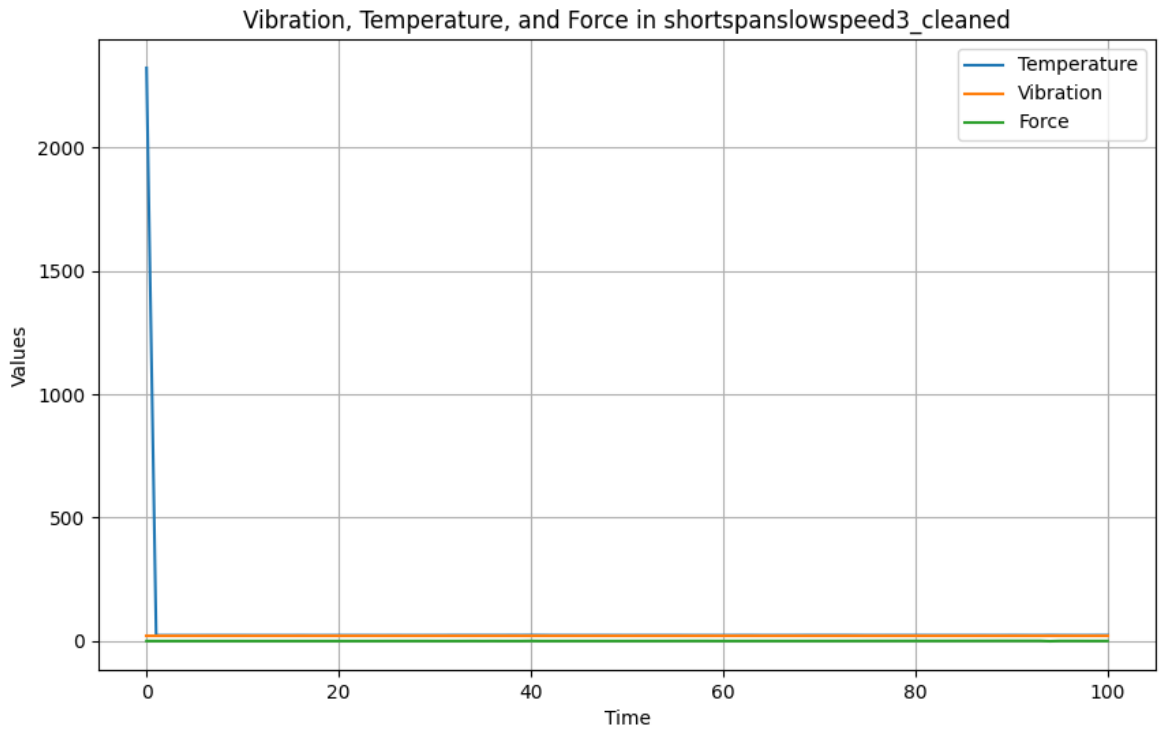


Figure 47: Vibration, Temperature, force in short span slow drill speed iteration 3

4.3 DATA COMPACTION

To obtain the data to be used for analysis, the averages of the iterations were first obtained. This reduced the graphs from 18 to 6.

The resulting average graphs:

4.3.1 Long span action (Average Vibration, Temperature, Force in long span fast drill speed)

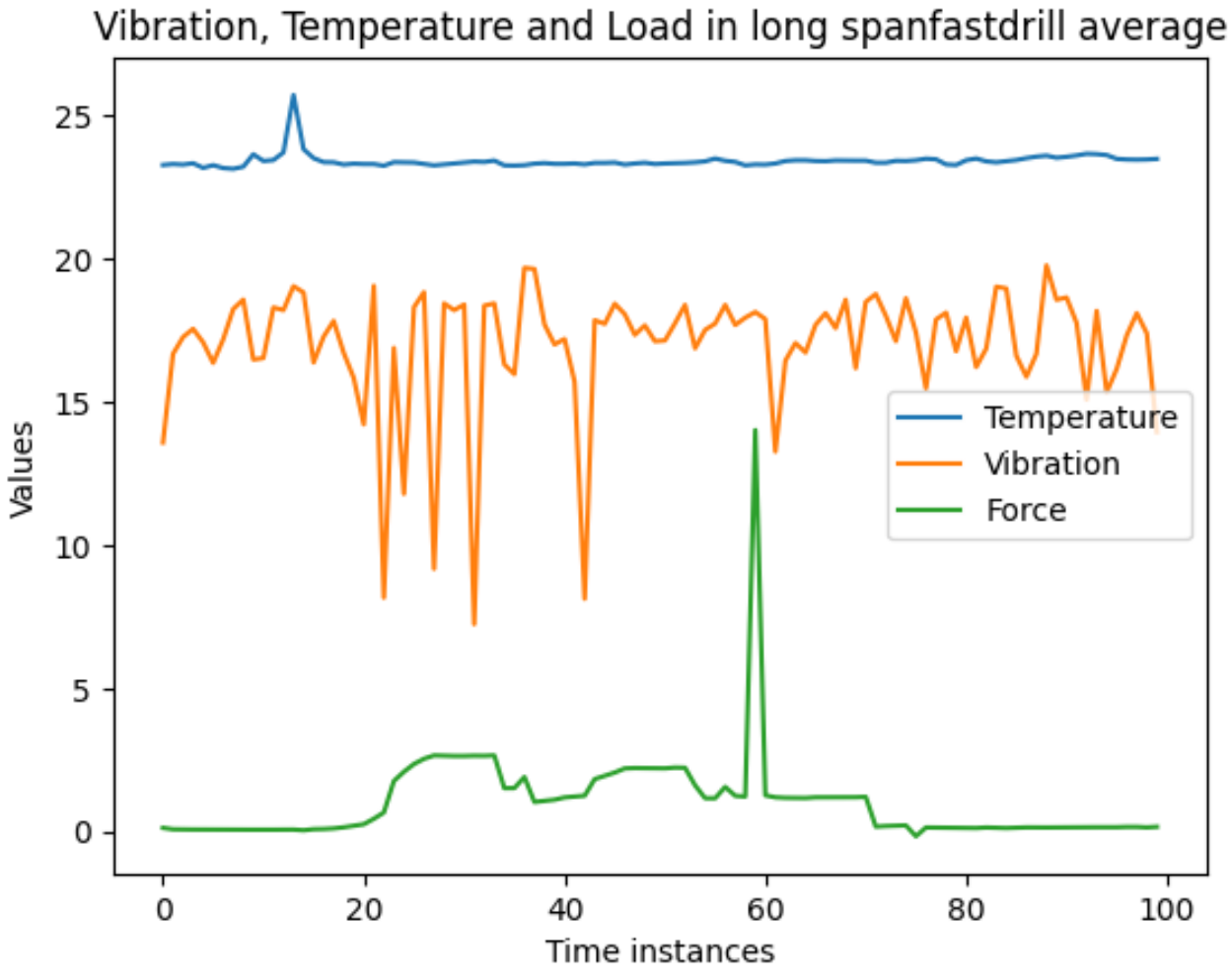


Figure 48: Average Vibration, Temperature, Force in long span fast drill speed

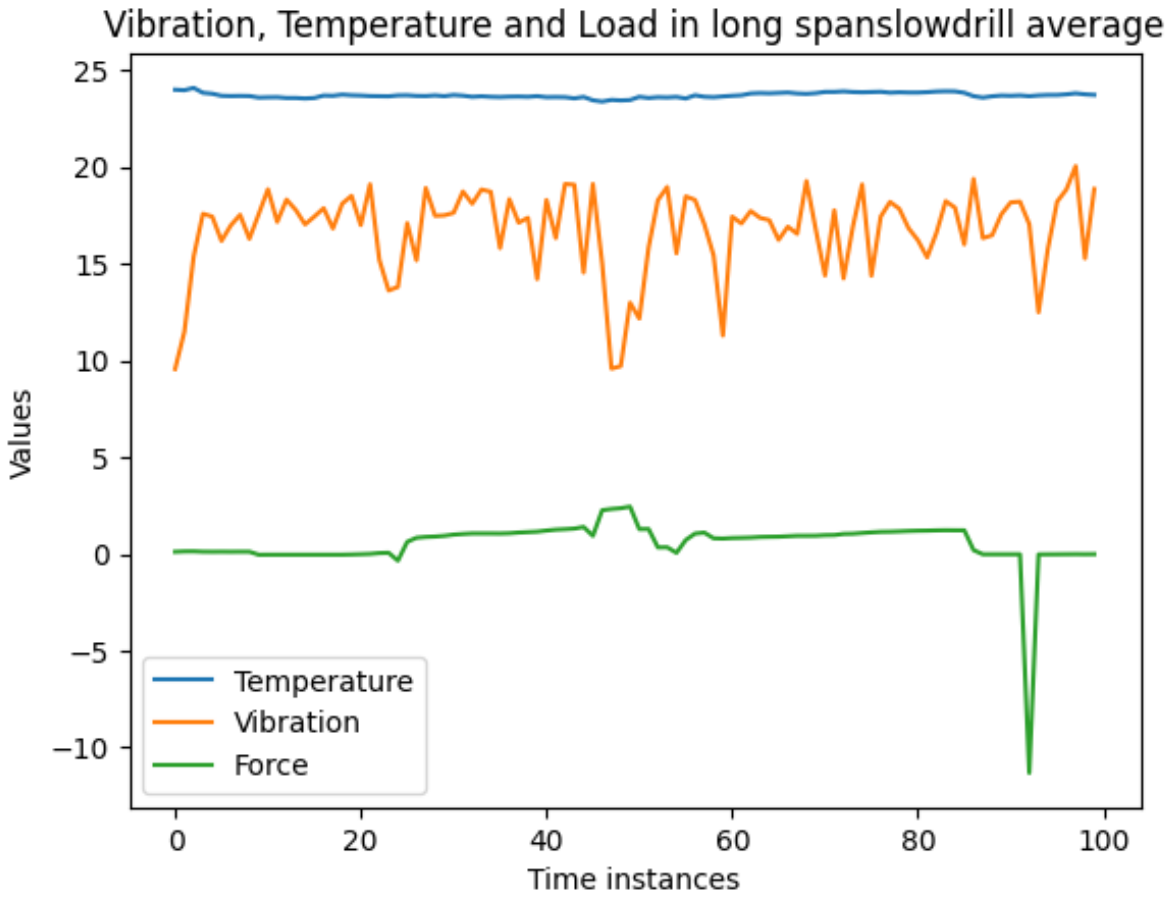


Figure 49: Average Vibration, Temperature, Force in long span slow drill speed

4.3.2 Medium span action (Average Vibration, Temperature, Force in medium span fast drill speed)

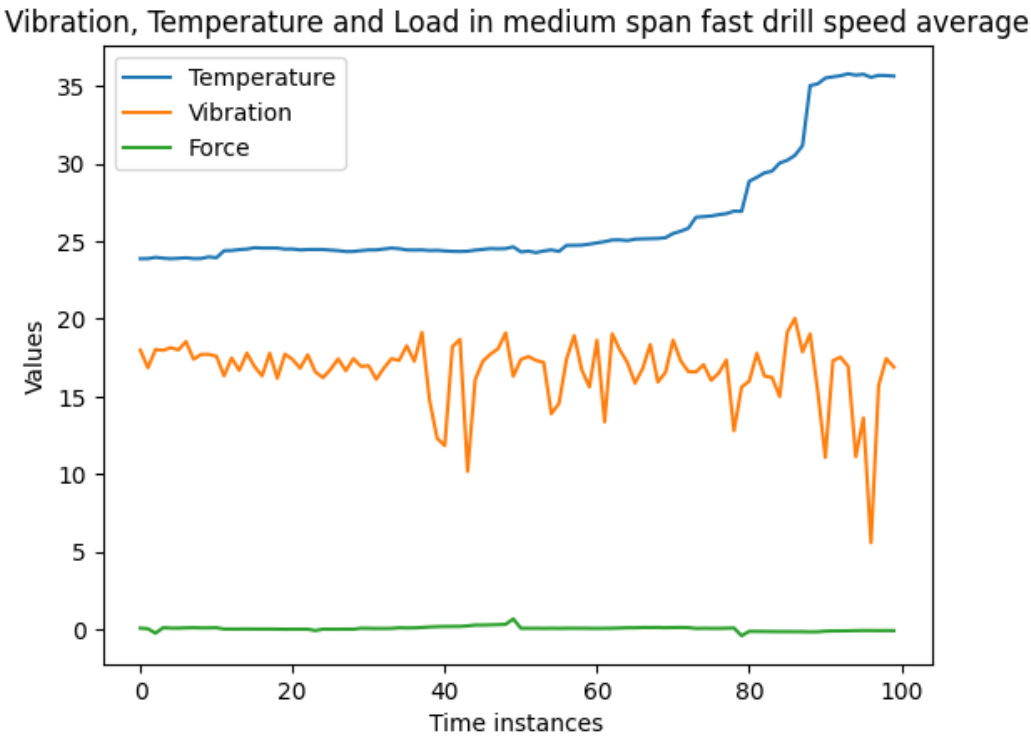


Figure 50: Average Vibration, Temperature, Force in medium span fast drill speed

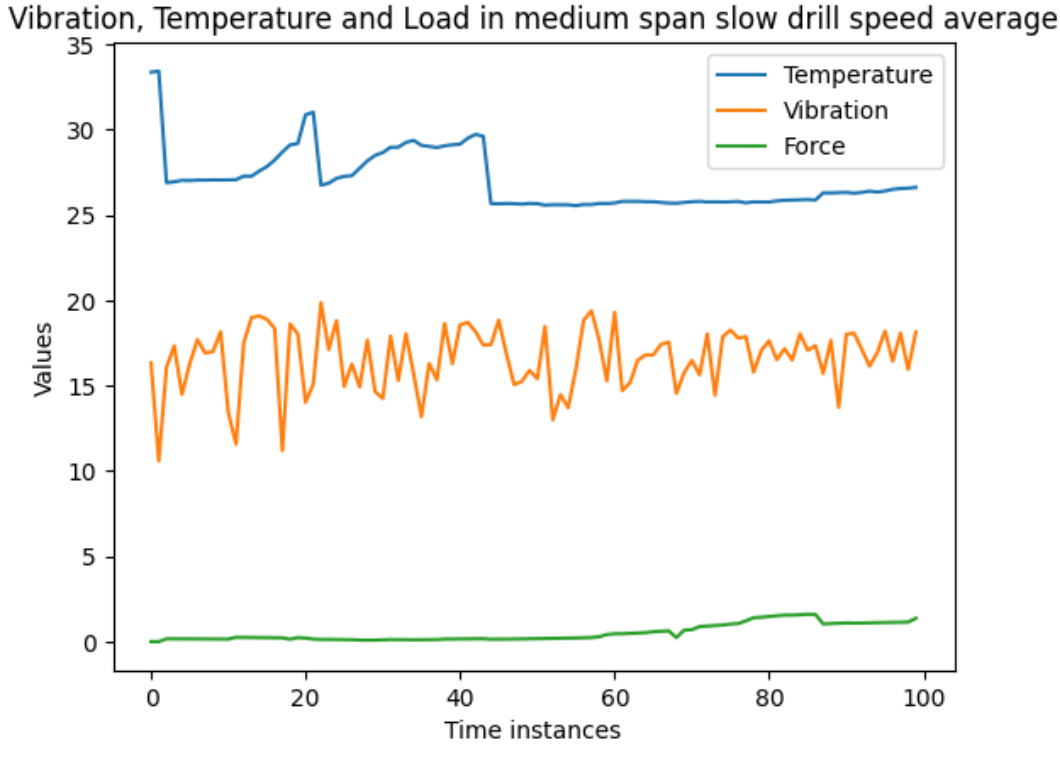


Figure 51: Average Vibration, Temperature, Force in medium span slow drill speed

4.3.2 Short span action (Average Vibration, Temperature, Force in short span fast drill speed)

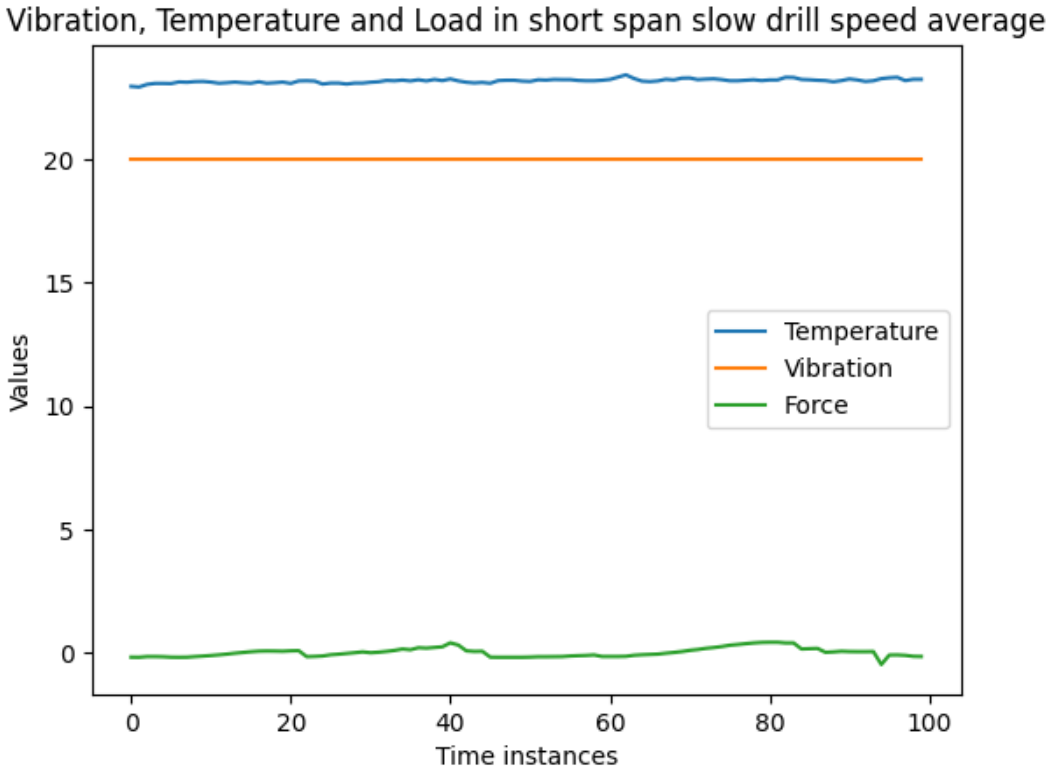


Figure 52: Average Vibration, Temperature, Force in short span slow drill speed

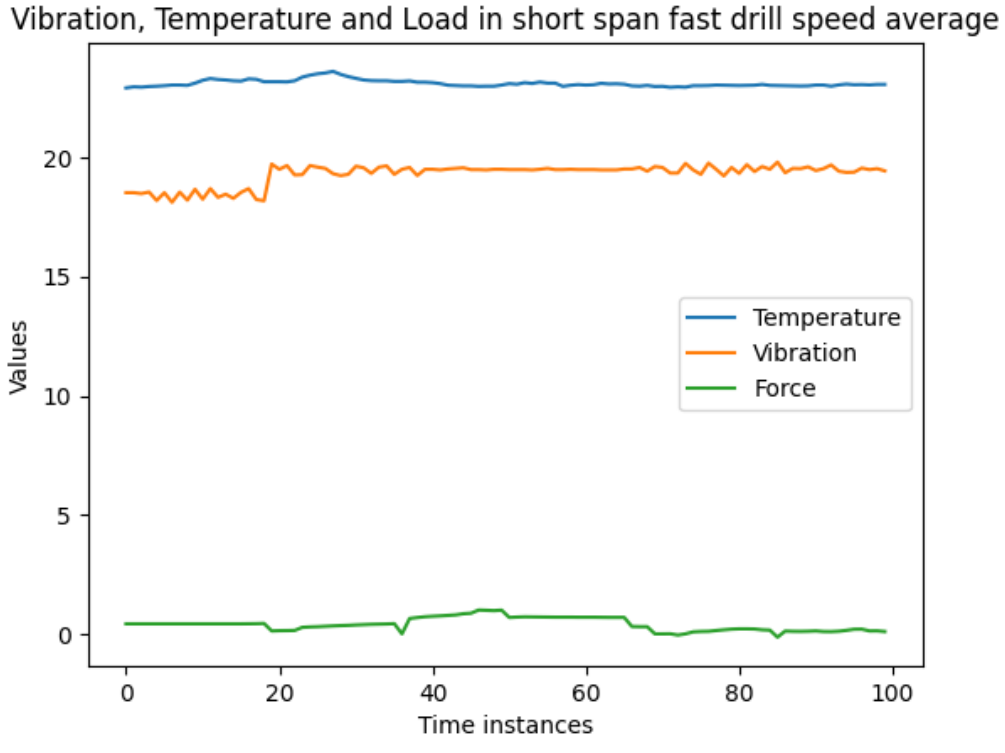


Figure 53: Average Vibration, Temperature, Force in short span fast drill speed

4.4 DATA ANALYSIS

4.4.1 SPEARMAN RANK CORRELATION COEFFICIENT

To determine the level of correlation between the variables

4.4.1.a Long span stand, fast drill speed action

Measured both vibration and load applied only.
Sample Data and Correlation Analysis between force and vibration

Table 3: Sample Data and Correlation Analysis between force and vibration

Vibration (acceleration in m/s^2)	Force/ Load applied (Kg)	Rank Vibration	Rank Force
13.58666667	0.126666667	7	21
16.68333333	0.076666667	27	15
17.27333333	0.07	43	12.5
17.56	0.07	50	12
17.08666667	0.07	37	11.5
16.37	0.07	21	11
17.19666667	0.063333333	41	6
18.25333333	0.063333333	75	5.5
18.58	0.063333333	87.5	5
16.48666667	0.063333333	24	4.5

The findings of rank correlation calculation:

Table 4: The findings of rank correlation calculation:

correlation between force and vibration	
Correlation coefficient®	0.039286142
Number of pairs(N)	100
T- statistic)	0.389213434
DF	98
P- value	0.697962714

The p- value is >0.05; this means there is not enough evidence to reject null hypothesis, therefore null hypothesis remains valid.
Null hypothesis: There is no significant relationship between vibration and force readings.

4.4.1.b Medium span, fast drill speed action.

The measured variables of significance were vibration, force and temperature
Sample Data and Correlation analysis between vibration and temperature:

Table 5: Sample Data and Correlation analysis between vibration and temperature:

Temperature (degree Celsius)	Vibration (acceleration in m/s^2)	Force/load applied (Kgs)	Temperature Rank	Vibration rank	Force Rank
23.88333	17.99	0.063333	1.5	81	65.5
23.89	16.85333	0.023333	3.5	43	41
23.97	18.03	-0.26	10	84	2
23.91667	17.97667	0.093333	7	80	79.5
23.88333	18.14667	0.066667	1.5	86	67
23.91	18.00333	0.063333	6	82.5	65.5
23.94333	18.54	0.083333	8.5	90	73.5
23.89	17.40333	0.093333	3.5	61	79.5
23.89667	17.71333	0.08	5	72	71.5
24.00333	17.71667	0.08	11	73	71.5

Table 6: Correlation between vibration and temperature

Correlation between vibration and temperature	
Correlation coefficient	-0.22962312
Number of pairs	100
T- statistic	2.335559958
Degrees of freedom	98
P- value	0.02155263

The correlation coefficient is negative.
The p value<0.05; this means there is enough evidence to reject the null hypothesis.
Null hypothesis: There is no significant relationship between vibration and temperature.
There is a significant relationship between vibration and temperature.
Correlation between vibration and force

Table 7: Correlation between vibration and force

Correlation between vibration and force	
Correlation coefficient	0.129562604
Number of pairs	100
T- statistic	1.293506998
Degrees of freedom	98

P- Value	0.198876329

The p value is >0.05, thus not enough evidence to reject the null hypothesis.
There is no significant relationship between vibration and force.
Correlation between force and temperature

Table 8: Correlation between force and temperature

Correlation between force and temperature	
Correlation coefficient	-0.441824124
Number of pairs	100
T- statistic	4.875518268
Degrees of freedom	98
P- Value	4.18864E-06

There is negative correlation.
The p value<0.05, thus there is sufficient evidence to reject null hypothesis; there is a significant relationship between force and temperature.

4.4.1 DESCRIPTIVE STATISTICS

4.4.2.a long span fast drill speed action

Table 9: DESCRIPTIVE STATISTICS (vibration)

DESCRIPTIVE STATISTICS (vibration)	
Mean	16.9994
Range	12.53666667
Variance	5.112525114
Standard deviation	2.261089364

4.4.2.b Medium span fast drill speed action

Table 10: DESCRIPTIVE STATISTICS (vibration)

DESCRIPTIVE STATISTICS (vibration)	
Mean	16.6576
Range	14.47
Variance	4.41495984
Standard deviation	2.101180582

4.4.3.c Short span fast drill speed action

Table 11: Descriptive statistics short span fast drill speed action

DESCRIPTIVE STATISTICS (vibration)	
Mean	19.28216667
Range	1.68333333
Variance	0.197906481
Standard deviation	0.444866813

Short span fast drill speed action has the lowest value of vibration, this is depicted by the smallest value of standard deviation, thus the lowest dispersion.
Long span fast drill speed action has the highest value of vibration, this is depicted by the highest value of standard deviation, thus the highest dispersion.

4.4.3 ANOVA Analysis

Objective: Determine if there are significant differences in temperature, vibration, and force at different PWM levels.
Since all sensor data was collected in medium span drilling action for both fast and slow drill speeds, Analysis of variance is carried out for these two iterations.

4.4.3.1(a) Medium span fast drill speed action

Sample data:

Table 12: Sample data for Anova analysis medium span fast drill speed action

Temperature (degree Celsius)	Vibration (acceleration in m/s^2	Force/load applied in Kgs	PWM Value
23.88333	17.99	0.063333	255
23.89	16.85333	0.023333	255
23.97	18.03	-0.26	255
23.91667	17.97667	0.093333	255
23.88333	18.14667	0.066667	255
23.91	18.00333	0.063333	255
23.94333	18.54	0.083333	255
23.89	17.40333	0.093333	255
23.89667	17.71333	0.08	255
24.00333	17.71667	0.08	255

ANOVA Results:

Table 13: ANOVA Results for ANOVA analysis medium span fast drill speed action:

Anova: Single Factor						
SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
Temp.av	100	2643.7	26.437	13.94582		
Vib.av	100	1665.76	16.6576	4.41496		
Force.av	100	3.046667	0.030467	0.018092		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	35646.77	2	17823.39	2909.328	8.3E-196	3.026153
Within Groups	1819.508	297	6.126291			
Total	37466.28	299				

The P-value for the between groups variation is very close to zero (8.2708E-196), which is less than the significance level (commonly set at 0.05). This indicates a significant difference among the groups for all three variables (Temperature, Vibration, Force).
The F-statistic is very high, further supporting the evidence of significant differences.

4.4.3.1(b) Medium span slow drill speed action

Sample data:

Table 14: Sample data for Anova analysis medium span slow drill speed action

Temperature (degree Celsius)	Vibration (acceleration in m/s^2)	Force/ Load applied in Kgs	PWM Value
33.35	16.33	0.003333	200
33.41	10.58	0.003333	200
26.87667	16.07	0.166667	200
26.93	17.32667	0.16	200
27.01667	14.47667	0.163333	200
27.00333	16.26333	0.16	200
27.03	17.69	0.15	200
27.03	16.90667	0.143333	200
27.04333	16.98333	0.143333	200
27.03667	18.15	0.136667	200

ANOVA Results:

Table 15: Sample results for Anova analysis medium span slow drill speed action

Anova: Single Factor						
SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
Temp.av	100	2700.287	27.00287	2.766562		
Vib.av	100	1658.537	16.58537	3.357169		
Force.av	100	51.00667	0.510067	0.23873		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	35626.93	2	17813.47	8399.329	4.2E-262	3.026153
Within Groups	629.8836	297	2.12082			
Total	36256.82	299				

Similar to the fast drill speed, the P-value for the between groups variation is very close to zero (4.1763E-262), indicating a significant difference among the groups for all three variables.
The F-statistic is extremely high, providing strong evidence of significant differences.

4.4.3.2 Further analysis of variance(anova)

1. **Data collection**

The goal of this analysis is to obtain optimal operating points for the stepper motor which provides linear feed to the dc motor station and the dc motor speed which provides rotary feed to the workpiece.

The two hypotheses in the experiment:

Null hypothesis: There is no significant difference in the mean quality of drilling across different levels of linear feed speed provided by the stepper motor and rotary speed determined by the DC motor.

Alternative hypothesis: There is a significant difference in the mean quality of drilling across different levels of linear feed speed provided by the stepper motor and rotary speed determined by the DC motor.

Table 16: Data Collection and analysis Anova

	Stepper Motor Speed (RPM)	DC Motor Speed (RPM)	Drill performance
E0	26.7	7958	0
E1	30.0	9422	0
E2	24.2	9422	0
E3	17.4	9604	3
E4	17.8	9775	3
E5	17.4	9712	2
E6	9.8	9778	8
E7	9.9	9671	5

E8	9.4	9733	8
E9	31.1	9785	5
E10	21.4	9836	5
E11	4.5	9706	3

2. ANOVA analysis

Drill performance extremes: 0- does not drill

10- drills efficiently

Table 17: Anova analysis

ANOVA: Two-Factor Without Replication

SUMMARY	*Count*	*Sum*	*Average*	*Variance*
E0	2	7984.7	3992.35	31452760
E1	2	9452	4726	44104832
E2	2	9446.2	4723.1	44159322
E3	2	9621.4	4810.7	45951450
E4	2	9792.8	4896.4	47601476
E5	2	9729.4	4864.7	46992635
E6	2	9787.8	4893.9	47708866
E7	2	9680.9	4840.45	46668427
E8	2	9742.4	4871.2	47274198
E9	2	9816.1	4908.05	47569283
E10	2	9857.4	4928.7	48163187
E11	2	9710.5	4855.25	47059551
Stepper Motor Speed (RPM)	12	219.6	18.3	75.45818
DC Motor Speed (RPM)	12	114402	9533.5	264111

ANOVA

Table 18: Anova Sources of Variance

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Stepper*DC Motor	1434252	11	130386.5	0.974489	0.516708	2.81793

Stepper and DC Motor Separately	5.43E+08	1	5.43E+08	4060.048	1.76E-15	4.844336
Error	1471799	11	133799.9			
Total	5.46E+08	23				
Total	7.28E+08	35				

3. Interpretation of ANOVA results

- The p-values indicate the significance of each factor and their interactions.
- Stepper Motor Speed has a p-value of <0.05, indicating that it has a significant effect on Optimal Drilling Performance.
- Drill Motor Speed also has a p-value of <0.05, suggesting a significant effect on Optimal Drilling Performance.
- Both the stepper motor speed and the drill motor speed independently significantly impact the quality or efficiency of the drilling process for PMMA material.
- The interaction between Stepper Motor Speed and Drill Motor Speed has a p-value of 0.516708, which is not statistically significant. Therefore, the interaction does not have a significant effect on Optimal Drilling Performance.
- The p- value of interaction between stepper motor speed and dc motor speed is greater than 0.5, thus we fail to reject null hypothesis; this means there is no significant difference in the mean quality of drilling across different levels of linear feed speed provided by the stepper motor and rotary speed determined by the DC motor. This means that the two variables can be controlled independent of each other to obtain best quality drilling.

From the analysis, the number of times the station is able to carry out a successful drilling operation is limited. The station exhibits a successful drilling operation when a starting hole is provided. Failure to drill can be attributed to the melting of PMMA and the sticking of the melted PMMA on the drill bit, making further drilling difficult. The new PMMA used melts when heated during heating. Using the current setup as it is, the optimal operating parameters is at 200 pulses/rev setting for stepper motor, where it operates at 21.6 rpm and for the dc motor, at 255PWM, which is equivalent to 9778rpm, which can be conditioned separately, with one of the variables held constant.

4.4.3.3 ANOVA Analysis Conclusion:

For both fast and slow drill speeds, the ANOVA results suggest that there are significant differences among the groups (PWM values) for all three variables: temperature, vibration, and force.
The high F-statistics and very low P-values indicate strong evidence against the null hypothesis of equal means.

4.4.3.4 ANOVA Recommendation:

Using a geared dc motor for the drill station with operating speed limit of 300 rpm will provide sufficient torque to drill the PMMA as it melts. Temperature should also be control variable in the system due to thermal behavior of the PMMA strip. Based on these results, you can conclude that there are significant differences in temperature, vibration, and force at different drilling motor speeds represented by different PWM values.

4.5 FEA calculations

4.5.1 Mathematical analysis of pmma.

This analysis assumes the maximum force exerted on PMMA strip is the maximum rated load of the load cell we shall be using for the project: 10kg rated.
Considerations of experiment: dc motor running at 9600 rpm.
Drill bit material:

- high speed steel.
- Drill bit diameter: 4mm

Mechanical properties of PMMA:

- Poisson's ratio= 0.35
- Young's modulus=2.7 GPa
- Dimensions of PMMA strip: 180mm by 20mm by 4mm

Given that the force acting on the PMMA strip is 10kg, we convert this into newtons;

$$\begin{aligned} F &= m * g \\ &= 10Kg * 9.81m/s^2 \\ &= 98.1N \end{aligned}$$

The area of the circular part at the center of the PMMA strip is given by;

$$A = \frac{\pi d^2}{4} = \frac{\pi * (4mm)^2}{4} = 4\pi mm^2$$

Where d id the diameter of the circular part.

The stress caused by the force acting on the PMMA strip is given by;

$$\sigma = \frac{F}{A} = \frac{98.1N}{4\pi mm^2} \approx 7.83MPa$$

To calculate the maximum and minimum equivalent stresses, we need;

$$\sigma_{eq} = \frac{\sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}}{2}$$

Where $\sigma_1, \sigma_2 \; and \; \sigma_3$ are principal stresses

The maximum and minimum principal stresses are given by;

$$\sigma_{max/min} = \frac{\sigma_1 + \sigma_2}{2} \pm \sqrt{(\frac{\sigma_1 - \sigma_2}{2})^2 + {\sigma_3}^2}$$

To calculate the principle stresses, we use;

$$\sigma_1 = \frac{\sigma_x + \sigma_y}{2} + \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2}$$

$$\sigma_2 = \frac{\sigma_x + \sigma_y}{2} - \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2}$$

Where $\sigma_x, \sigma_y \; and \; \tau_{xy}$ are normal and shear stresses on a plane Θ to the x- axis given by;

$$\sigma_x = \frac{\sigma_{xx} + \sigma_{yy}}{2} + \frac{\sigma_{xx} - \sigma_{yy}}{2} cos(2\theta) + \tau_{xy} \sin(2\theta)$$

$$\sigma_y = \frac{\sigma_{xx} + \sigma_{yy}}{2} + \frac{\sigma_{xx} - \sigma_{yy}}{2} cos(2\theta) - \tau_{xy} \sin(2\theta)$$

$$\tau_{xy} = \frac{F}{A} * \frac{d}{4}$$

Where $\sigma_{xx} and \; \sigma_{yy}$ are the normal stresses in the x and y directions respectively.

Using the given dimensions of the PMMA strip, we can calculate the normal stresses in the x and y directions as follows:

$$\sigma_{xx} = \frac{F}{wt} = \frac{98.1N}{(20mm)(4mm)} \approx 1.3MPa$$

$$\sigma_{yy} = \frac{F}{lt} = \frac{98.1N}{(180mm)(4mm)} \approx 0.14MPa$$

where w and l are the width and length of the PMMA strip, respectively, and t is the thickness of the PMMA strip.

Since Θ, the angle between the plane and the x- axis, is 0°;

$$\sigma_x = \frac{\sigma_{xx} + \sigma_{yy}}{2} + \frac{\sigma_{xx} - \sigma_{yy}}{2} cos(2\theta) + \tau_{xy} \sin(2\theta) \approx 1.23MPa$$

$$\sigma_y = \frac{\sigma_{xx} + \sigma_{yy}}{2} + \frac{\sigma_{xx} - \sigma_{yy}}{2} cos(2\theta) - \tau_{xy} \sin(2\theta) \approx 0.14MPa$$

We can therefor calculate the principle stresses as;

$$\sigma_1 = \frac{\sigma_x + \sigma_y}{2} + \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2} \approx 1.23MPa$$

$$\sigma_2 = \frac{\sigma_x+\sigma_y}{2} - \sqrt{\left(\frac{\sigma_x-\sigma_y}{2}\right)^2 + \tau_{xy}^2} \approx -0.95MPa$$

In this case, the PMMA strip is subjected to a force acting on a circular point at its center. Since the force is applied at the center, the stress state is symmetric about the centerline of the strip. Therefore, the stress component σ_3 is zero.

$$\sigma_3 = 0$$

For the maximum and minimum equivalent stresses, we have;

$$\sigma_{eq,max} = \sqrt{\frac{(\sigma_1 - \sigma_2)^2 + \sigma_2^2 + \sigma_1^2}{2}} = 1.59MPa$$

$$\sigma_{eq,min} = 0$$

For the maximum and minimum principal stresses, we have;

$$\sigma_{max} = \frac{\sigma_1+\sigma_2}{2} + \sqrt{(\frac{\sigma_1 - \sigma_2}{2})^2 + \sigma_3^2} = 0.64MPa$$

$$\sigma_{min} = \frac{\sigma_1 + \sigma_2}{2} - \sqrt{\left(\frac{\sigma_1 - \sigma_2}{2}\right)^2 + \sigma_3^2} = -1.04MPa$$

Using the given values of Young's modulus, Poisson's ratio, and density of PMMA, we can calculate the shear modulus as follows:

$$G = \frac{E}{2(1+\nu)} = \frac{2.7GPa}{2(1+0.35)} \approx 0.96GPa$$

where E is the Young's modulus and ν is Poisson's ratio.

The shear strain can be calculated using the following equation:

$$\gamma_{xy} = \frac{\Delta x}{h}$$

where Δx is the displacement in the x direction and *h* is the thickness of the PMMA strip.

Assuming that the PMMA strip is in a state of plane stress, we can use the following equation to calculate the displacement in the x direction:

$$\Delta x = \frac{\sigma_{xx}}{E} * x$$

where x is the distance from the neutral axis of the PMMA strip.

Using the given dimensions of the PMMA strip, we can calculate the distance from the neutral axis as follows:

$$x = \frac{t}{2} = 2mm$$

Using these values, we can calculate the shear strain as follows:

$$\gamma_{xy} = \frac{\Delta x}{h} = \frac{\sigma_{xx}}{Eh} * x = \frac{1.23MPa}{(2.7GPa)(4mm)} * 2mm \approx 0.23$$

From shear strain, the expected deflection of PMMA during drilling:

$$\Delta x = \gamma_{xy} * h = 0.23 * 4 = 0.92\ mm$$

The results of this analytical calculations are compared with the results of finite element analysis in Abaqus and Ansys to determine best drilling parameters.

Chapter 5: Conclusion and Recommendations

5.1 Conclusion

In conclusion, the ANOVA analysis of drilling performance with PMMA material has provided valuable insights into the factors that influence the quality and efficiency of drilling operations. The findings reveal that both Stepper Motor Speed and Drill Motor Speed have a significant and independent impact on Optimal Drilling Performance. This highlights the critical role that motor speed adjustments play in achieving the desired drilling outcomes when working with PMMA material.

Furthermore, the lack of statistically significant interaction effects between Stepper Motor Speed and Drill Motor Speed, as well as between Stepper Motor Speed and DC Motor Speed, suggests that these variables can be controlled independently. This flexibility in motor speed control offers opportunities for fine-tuning the drilling process to meet specific requirements without compromising drilling qualityIn summary, the ANOVA analysis serves as a foundation for enhancing drilling performance with PMMA material. It empowers operators and engineers to make informed decisions regarding motor speed adjustments and highlights the potential for achieving consistent and high-quality drilling results. The subsequent implementation of a new DC motor, advanced control methods, and digital tuning techniques, as recommended, can further refine and optimize the drilling process, leading to improved efficiency and reliability.

5.2 Recommendations

- New DC Motor : Selecting a new DC motor with improved specifications and precision can have a significant impact on drilling performance. Consider a motor that offers higher torque and speed control capabilities. Ensure that the motor is compatible with the drilling station's requirements and can achieve the desired RPM (revolutions per minute). A motor with precise RPM control and low vibration can contribute to smoother and more consistent drilling.
- Control Method :
- Implement a more advanced control method, such as a closed-loop control system. This involves using feedback mechanisms, such as encoders or sensors, to continuously monitor the motor's performance and adjust its speed and position in real-time. Closed-loop control systems provide greater accuracy and stability, which is crucial for drilling with PMMA material. Additionally, this approach allows for adaptive control, where the system can automatically adjust motor speed based on the drilling conditions and material properties.
- Digital Tuning : Utilize digital tuning techniques to optimize the motor's performance parameters. Digital tuning involves adjusting control parameters through software interfaces to achieve the desired drilling outcomes. Parameters such as PID (Proportional-Integral-Derivative) gains can be fine-tuned digitally to optimize motor speed and position control. Digital tuning allows for precise adjustments and quick response to changing drilling conditions.
- Integration : Ensure seamless integration of the new DC motor and control method into the existing drilling station. This may require modifications to the hardware and software interfaces of the station to accommodate the new components. Calibration and testing should be conducted to verify that the motor and control system are operating as intended.
- Data Analysis : Utilize data analysis techniques to identify patterns and trends in drilling performance. This includes analyzing data from sensors, motor performance logs, and drilling

outcomes. Machine learning algorithms can be employed to predict optimal motor settings based on historical data and current conditions.

By incorporating a new DC motor, advanced control methods, and digital tuning into the drilling station, it is possible to achieve finer control over the drilling process. This approach can lead to increased precision, reduced drilling failures, and improved overall efficiency when working with PMMA material. However, thorough testing and validation are essential to ensure that the new components and control strategies effectively meet the desired drilling objectives.

References

Automated drilling. (2015, August 4). PetroWiki. https://petrowiki.spe.org/Automated_drilling

Betsaida Lo-Amni Fernandez Difo. (2016).´ *Automated Calibration for Numerical Models of Riverflow* (Unpublished doctoral dissertation). University of Stuttgart - Institute for Modelling Hydraulic and Environmental Systems.

Cayeux, E., Daireaux, B., Ambrus, A., Mihai, R., & Carlsen, L. (2021). Autonomous Decision-Making While Drilling. *Energies*, *14*(4), 969. https://doi.org/10.3390/en14040969

Downing, D. J., Gardner, R. H., & Hoffman, F. O. (1985, May). An Examination of Response-Surface Methodologies for Uncertainty Analysis in Assessment Models. *Technometrics*, *27*(2), 151–163. Retrieved 2020-08-10, from http://www.tandfonline.com/doi/abs/10.1080/ 00401706.1985.10488032 doi: 10.1080/00401706.1985.10488032

Du, G., Zhang, P., & Li, D. (2015, February). Online robot calibration based on hybrid sensors using Kalman Filters. *Robotics and Computer-Integrated Manufacturing*, *31*, 91–100. Retrieved 2020-04-22, from https://linkinghub.elsevier.com/ retrieve/pii/S0736584514000635 doi: 10.1016/j.rcim.2014.08.002

Elatta, A., Zhi, F. L., Daoyuan, Y., Fei, L., & Pei Gen, L. (2004, January). An Overview of Robot Calibration. *Information Technology Journal*, *3*(1), 74–78. Retrieved 2020-04-22, from http://www.scialert.net/abstract/?doi= itj.2004.74.78 doi: 10.3923/itj.2004.74.78

Eustes, A. W. (2007, November 11). *The Evolution of Automation in Drilling*. Onepetro.org; OnePetro. https://doi.org/10.2118/111125-MS

Everything You Need To Know About Acrylic (PMMA). (n.d.). Www.xometry.com. https://www.xometry.com/resources/materials/acrylic-pmma/

E.M. Purcell. (1977). *Life at low Reynolds number.* Am. J. Phys.

Etievant, M., Bolopion, A., Regnier, S., & Andreff, N. (2019, May). An Improved Control-Oriented Modeling of the Magnetic Field. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 6178–6184). Montreal, QC, Canada: IEEE. Retrieved 2020-03-18, from https://ieeexplore.ieee.org/document/ 8793679/ doi: 10.1109/ICRA.2019.8793679

Gwinnett, A., & Gorelick, L. (1998). A Brief History of Drills and Drilling. *BEADS: Journal of the Society of Bead Researchers*, *10*(11), 1998. https://surface.syr.edu/cgi/viewcontent.cgi?article=1100&context=beads

Gardner, R., O'Neill, R., Mankin, J., & Carney, J. (1981, April). A comparison of sensitivity analysis and error analysis based on a stream ecosystem model. *Ecological Modelling*, *12*(3), 173–190. Retrieved 2020-08-10, from https://linkinghub .elsevier.com/retrieve/pii/0304380081900569 doi: 10.1016/0304 -3800(81)90056-9

Gauthier, M. (2013). *Intracorporeal robotics*. Hoboken, NJ: ISTE Ltd/John Wiley and Sons Inc.

Grant, I. S., & Phillips, W. R. (1990). *Electromagnetism* (2nd ed ed.). Chichester [England] ; New York: Wiley.

Hao, P., Yu, C., Feng, T., Zhang, Z., Qin, M., Zhao, X., He, H., & Yao, X. S. (2020). PM fiber based sensing tapes with automated 45° birefringence axis alignment for distributed force/pressure sensing. *Optics Express*, *28*(13), 18829–18842. https://doi.org/10.1364/OE.391376

Hamby, D. M. (1994, September). A review of techniques for parameter sensitivity analysis of environmental models. *Environmental Monitoring and Assessment*, *32*(2), 135–154. Retrieved 2020-05-19, from http://link.springer.com/10.1007/ BF00547132 doi: 10.1007/BF00547132

Iino, T. (1969, December). Genetics and chemistry of bacterial flagella. *Bacteriological Reviews*, *33*(4), 454–475. Retrieved 2020-04-15, from https://www.ncbi.nlm .nih.gov/pmc/articles/PMC378339/

Johnson, L. (2023, September 20). *Revolutionizing the Oil and Gas Industry: Automating the Drilling Process*. TOMORROW'S WORLD TODAY®. https://www.tomorrowsworldtoday.com/manufacturing/revolutionizing-the-oil-and-gas-industry-automating-the-drilling-process/

Jackson, J. D. (1999). *Classical electrodynamics* (3rd ed ed.). New York: Wiley.

Johansson, S. (1994, September). Techniques for the fabrication of micro-robot systems. *IFAC Proceedings Volumes*, *27*(14), 755–762. Retrieved 2020-

08-07, from https://linkinghub.elsevier.com/retrieve/pii/ S1474667017473939 doi: 10.1016/S1474-6670(17)47393-9

Khalil, I. S. M., Abelmann, L., & Misra, S. (2014, October). Magnetic-Based Motion Control of Paramagnetic Microparticles With Disturbance Compensation. *IEEE Transactions on Magnetics*, *50*(10), 1–10. Retrieved 2020-05-14, from http:// ieeexplore.ieee.org/document/6815708/ doi: 10.1109/TMAG.2014 .2323940

Kohidai, L. (2011). *Flagellum - beating pattern of flagellum and cilia.* Retrieved from https://en.wikipedia.org/wiki/File:Flagellum-beating.svg

Kolda, T. G., Lewis, R. M., & Torczon, V. (2003, January). Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *SIAM Review*, *45*(3), 385–482. Retrieved 2020-05-14, from http://epubs.siam.org/doi/10.1137/ S003614450242889 doi: 10.1137/S003614450242889

Kummer, M. P., Abbott, J. J., Kratochvil, B. E., Borer, R., Sengul, A., & Nelson, B. J. (2010, December). OctoMag: An Electromagnetic System for 5-DOF Wireless Micromanipulation. *IEEE Transactions on Robotics*, *26*(6), 1006–1017. Retrieved 2020-03-18, from http://ieeexplore.ieee.org/document/5595508/ doi: 10.1109/TRO.2010.2073030

Lagarias, J. C., Reeds, J. A., Wright, M. H., & Wright, P. E. (1998, January). Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions. *SIAM Journal on Optimization*, *9*(1), 112–147. Retrieved 2020-05-14, from http:// epubs.siam.org/doi/10.1137/S1052623496303470 doi: 10.1137/ S1052623496303470

Larson, J., Menickelly, M., & Wild, S. M. (2019, May). Derivative-free optimization methods. *Acta Numerica*, *28*, 287–404. Retrieved 202005-14, from https://www.cambridge.org/core/product/identifier/S0962492919000060/type/journalarticle doi: 10.1017/S0962492919000060

Lenshof, A., & Laurell, T. (2015). Acoustophoresis. In B. Bhushan (Ed.), *Encyclopedia of Nanotechnology* (pp. 1–6). Dordrecht: Springer Netherlands. Retrieved 2020-08-12, from http://link.springer.com/10.1007/978-94-007-6178-0423-2 doi: 10.1007/978-94-007-6178-0 423-2

Mack, M. J. (2001, February). Minimally Invasive and Robotic Surgery. *JAMA*, *285*(5), 568. Retrieved 2020-04-15, from http://jama.jamanetwork.com/ article.aspx?doi=10.1001/jama.285.5.568 doi: 10.1001/jama.285.5

.568

Marie, S., Courteille, E., & Maurine, P. (2013, November). Elasto-geometrical modeling and calibration of robot manipulators: Application to machining and forming applications. *Mechanism and Machine Theory*, *69*, 13– 43. Retrieved 2020-04-22, from https://linkinghub.elsevier.com/retrieve/pii/S0094114X13000967 doi: 10.1016/j.mechmachtheory.2013.05.003

Nocedal, J., & Wright ,Stephen J. (2006). *Numerical Optimization*. Springer New York. Retrieved 2020-08-11, from http://link.springer.com/10.1007/978-0 -387-40065-5 doi: 10.1007/978-0-

NJK. (n.d.). *Vol. 43, No. 1 (2020) – Journal of Mechanical Engineering Research and Developments*. https://jmerd.net/Paper/Vol.43

Postel, M., Candia, N., Bugdayci, B., Kuster, F., & Wegener, K. (2019). Development and application of an automated impulse hammer for improved analysis of five-axis CNC machine dynamics and enhanced stability chart prediction. *International Journal of Mechatronics and Manufacturing Systems*, *12*(3/4), 318. https://doi.org/10.1504/IJMMS.2019.103496

Petruska, A. J., Mahoney, A. W., & Abbott, J. J. (2014, October). Remote Manipulation With a Stationary Computer-Controlled Magnetic Dipole Source. *IEEE Transactions on Robotics*, *30*(5), 1222–1227. Retrieved 2020-05-14, from http:// ieeexplore.ieee.org/document/6871394/ doi: 10.1109/TRO.2014

.2340111

Ray, S., Mukherjee, J., & Mandal, S. (2015). Modelling nitrogen and carbon cycles in Hooghly estuary along with adjacent mangrove ecosystem. In *Developments in Environmental Modelling* (Vol. 27, pp. 289–320). Elsevier. Retrieved 2020-05-19, from https://linkinghub.elsevier.com/retrieve/pii/B9780444635365000132 doi: 10.1016/B978-0-444-63536-5.00013-2

Soleymani Eil Bakhtiari, S., Bakhsheshi-Rad, H. R., Karbasi, S., Tavakoli, M., Razzaghi, M., Ismail, A. F., RamaKrishna, S., & Berto, F. (2020). Polymethyl Methacrylate-Based Bone Cements Containing Carbon Nanotubes and Graphene Oxide: An Overview of Physical, Mechanical, and Biological Properties. *Polymers*, *12*(7), 1469. https://doi.org/10.3390/polym12071469

Soleymani Eil Bakhtiari, S., Bakhsheshi-Rad, H. R., Karbasi, S., Tavakoli, M., Hassanzadeh Tabrizi, S. A., Ismail, A. F., Seifalian, A., RamaKrishna, S., & Berto, F. (2020). Poly(methyl methacrylate)

bone cement, its rise, growth, downfall and future. *Polymer International*, *70*(9), 1182–1201. https://doi.org/10.1002/pi.6136

Siauve, N., Scorretti, R., Burais, N., Nicolas, L., & Nicolas, A. (2003, September). Electromagnetic fields and human body: A new challenge for the electromagnetic field computation. *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, *22*, 457–469. doi: 10.1108/ 03321640310474868

Sikorski, J., Dawson, I., Denasi, A., Hekman, E. E., & Misra, S. (2017, May). Introducing BigMag — A novel system for 3D magnetic actuation of flexible surgical manipulators. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3594–3599). Singapore, Singapore: IEEE. Retrieved 2020-07-15, from http://ieeexplore.ieee.org/document/7989413/ doi: 10.1109/ICRA.2017.7989413

Sikorski, J., Denasi, A., Bucchi, G., Scheggi, S., & Misra, S. (2019, April). Vision-Based 3-D Control of Magnetically Actuated Catheter Using BigMag—An Array of Mobile Electromagnetic Coils. *IEEE/ASME Transactions on Mechatronics*, *24*(2), 505–516. Retrieved 2020-07-15, from https://ieeexplore.ieee.org/document/ 8613884/ doi: 10.1109/TMECH.2019.2893166

Singer, S., & Nelder, J. (2009). Nelder-Mead algorithm. *Scholarpedia*, *4*(7), 2928. Retrieved 2020-05-14, from http://www.scholarpedia.org/article/ Nelder-Meadalgorithm doi: 10.4249/scholarpedia.2928

Teodoriu, C., & Bello, O. (2021). An Outlook of Drilling Technologies and Innovations: Present Status and Future Trends. *Energies*, *14*(15), 4499. https://doi.org/10.3390/en14154499

Tao, F., Zhang, L., & Laili, Y. (2015). *Configurable Intelligent Optimization Algorithm*. Cham: Springer International Publishing. Retrieved 2020-08-11, from http:// link.springer.com/10.1007/978-3-319-08840-2 doi: 10.1007/978-3 -319-08840-2

Tchon, K. (1992, October). Calibration of manipulator kinematics: a singularity theory approach. *IEEE Transactions on Robotics and Automation*, *8*(5), 671–678. Retrieved 2020-04-22, from http://ieeexplore.ieee.org/document/ 163792/ doi: 10.1109/70.163792

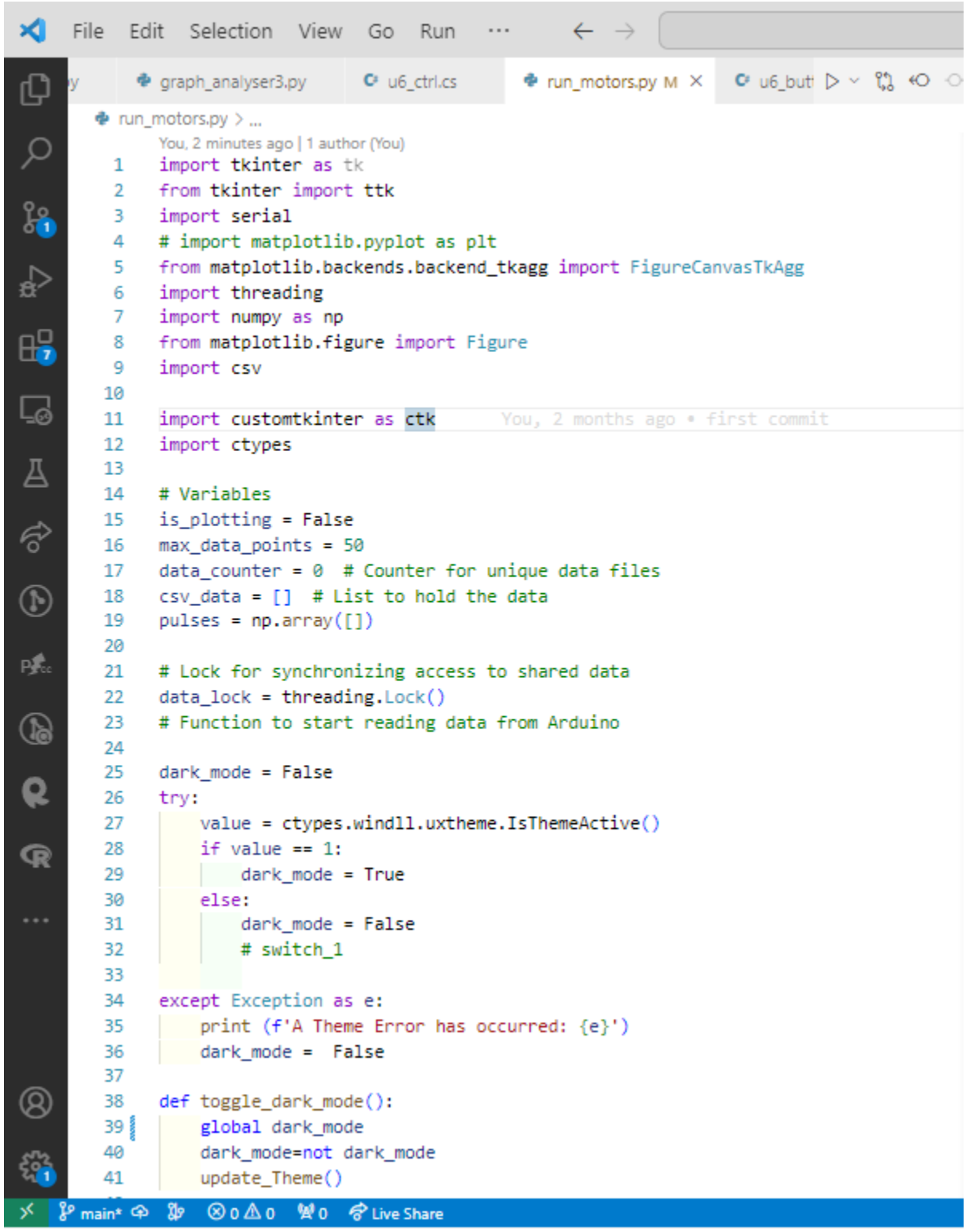
Tendick, F., Sastry, S., Fearing, R., & Cohn, M. (1998, March). Applications of micromechatronics in minimally invasive surgery. *IEEE/ASME Transactions on Mechatronics*, *3*(1), 34–42. Retrieved 2020-04-15, from http://ieeexplore.ieee .org/document/662866/ doi: 10.1109/3516.662866

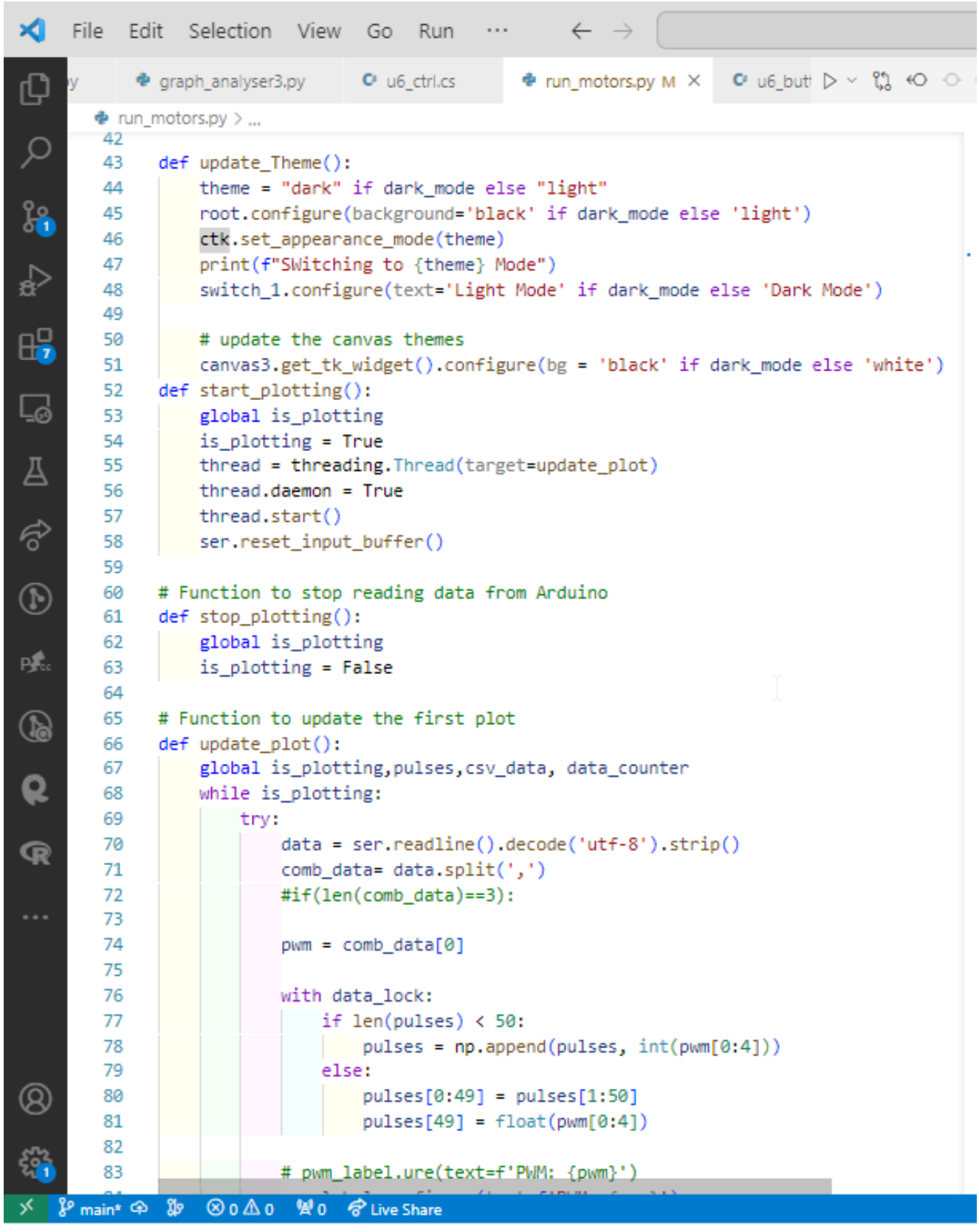
Vijini, Mallawaarachchi. (2017, July). *Towards data science.* Retrieved from https://towardsdatascience.com/introduction-to-genetic -algorithms-including-example-code-e396e98d8bf3

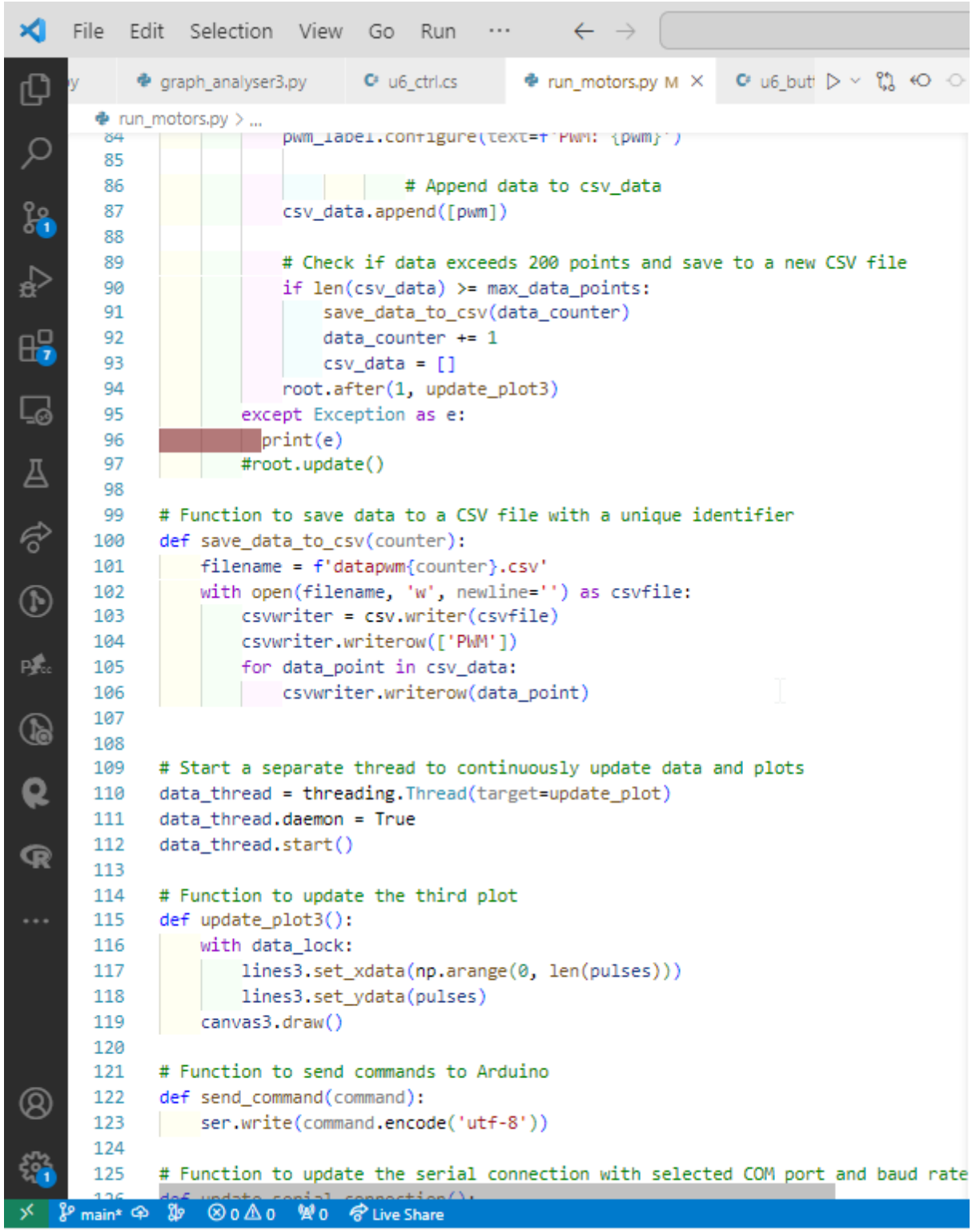
Wautelet, M. (2001, October). Scaling laws in the macro-, micro- and nanoworlds. *European Journal of Physics*, *22*(6), 601–611. Retrieved 2020-04-15, from https://doi .org/10.1088/0143-0807/22/6/305 (Publisher: IOP Publishing) doi: 10 .1088/0143-0807/22/6/305

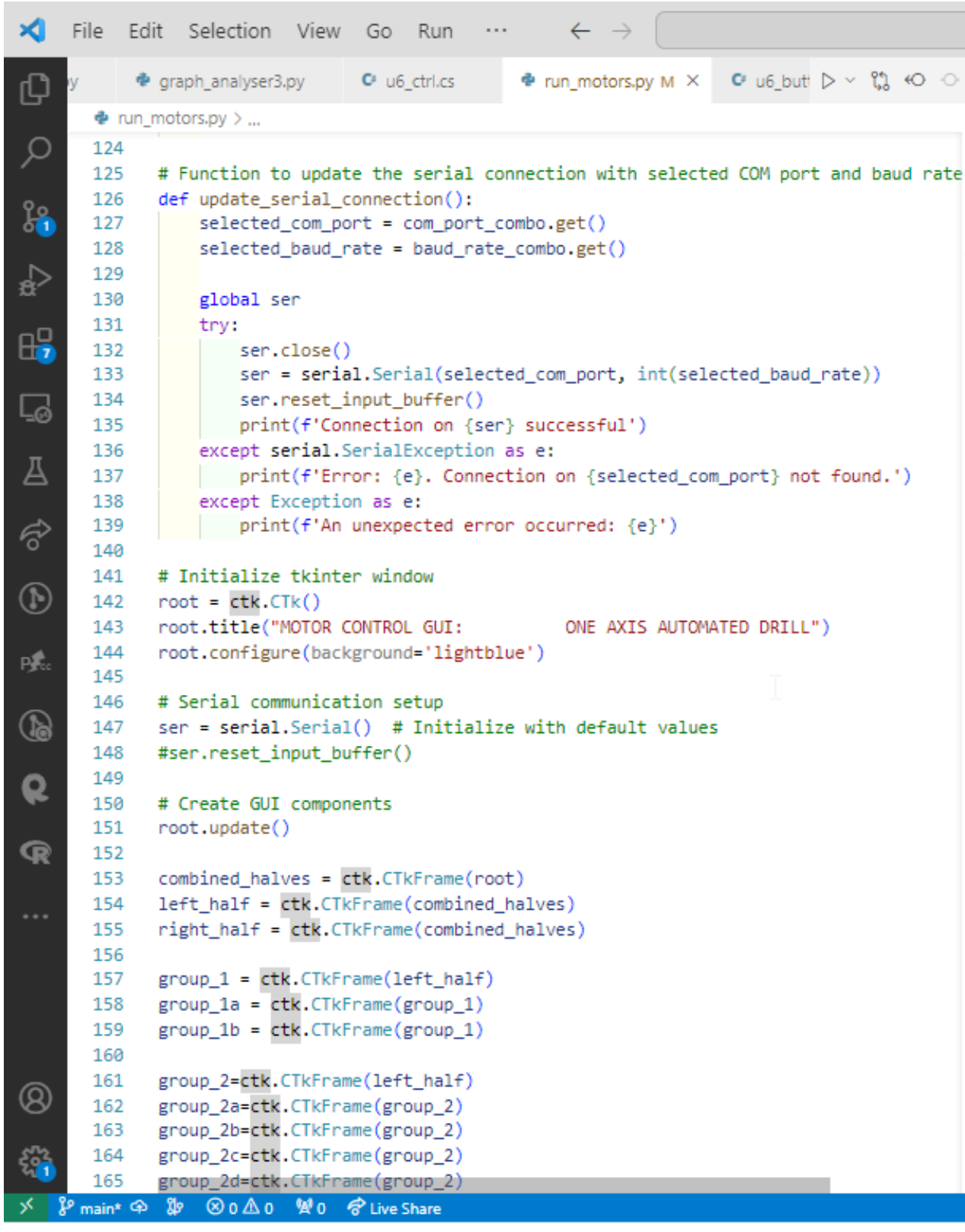
Appendix An Arduino source code

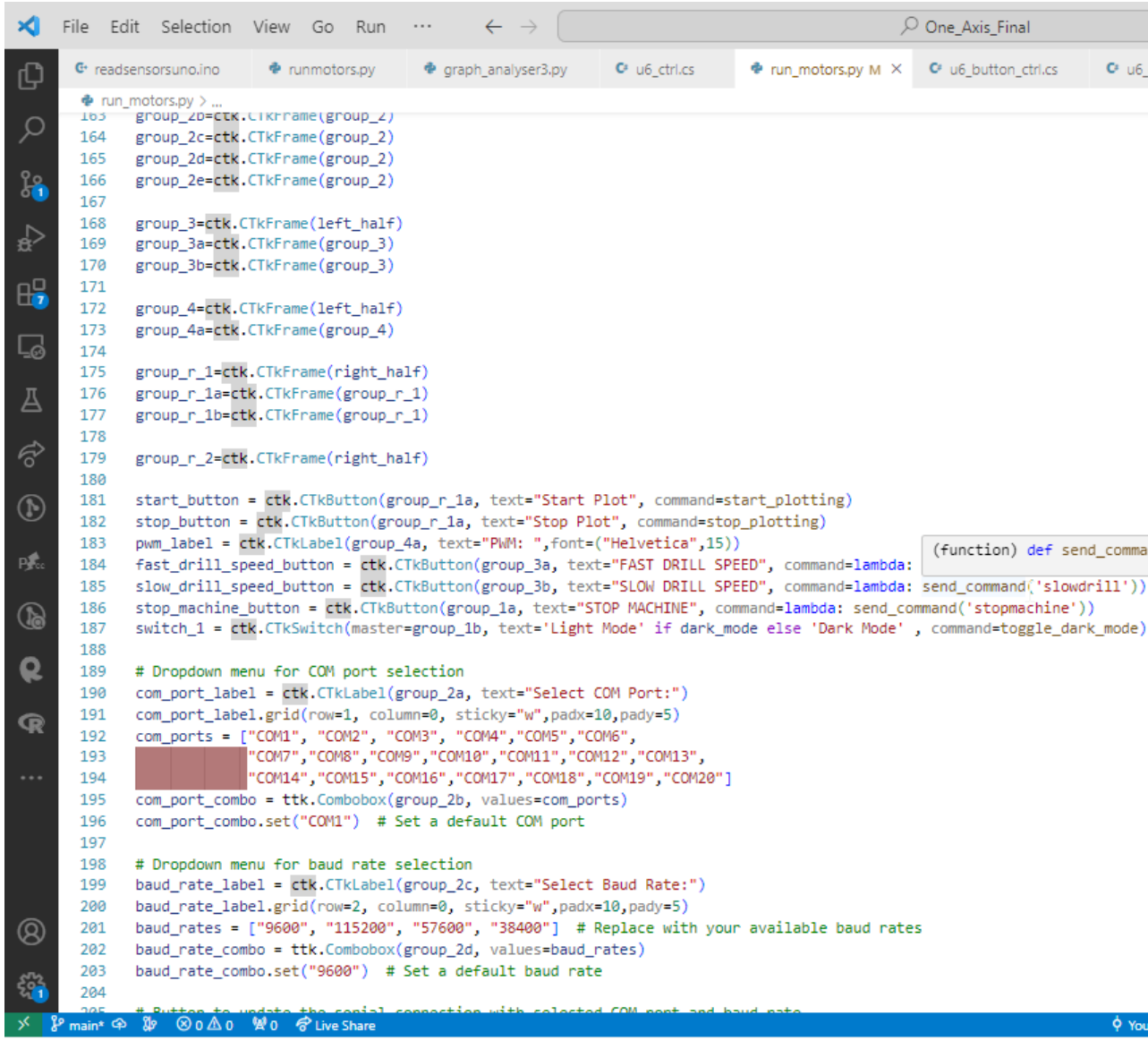
1) The code used to run the motors and provide with the run motor GUI

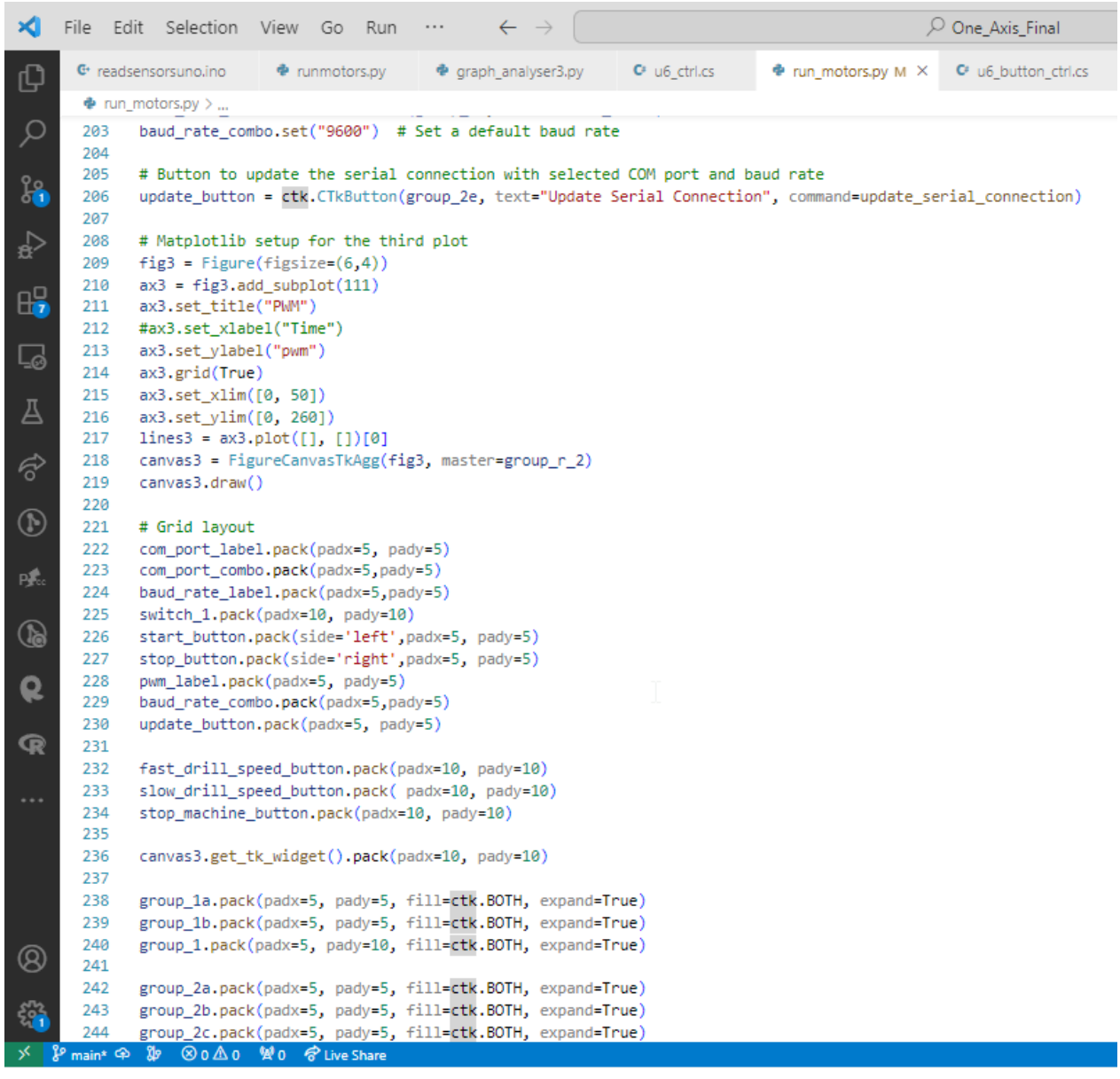


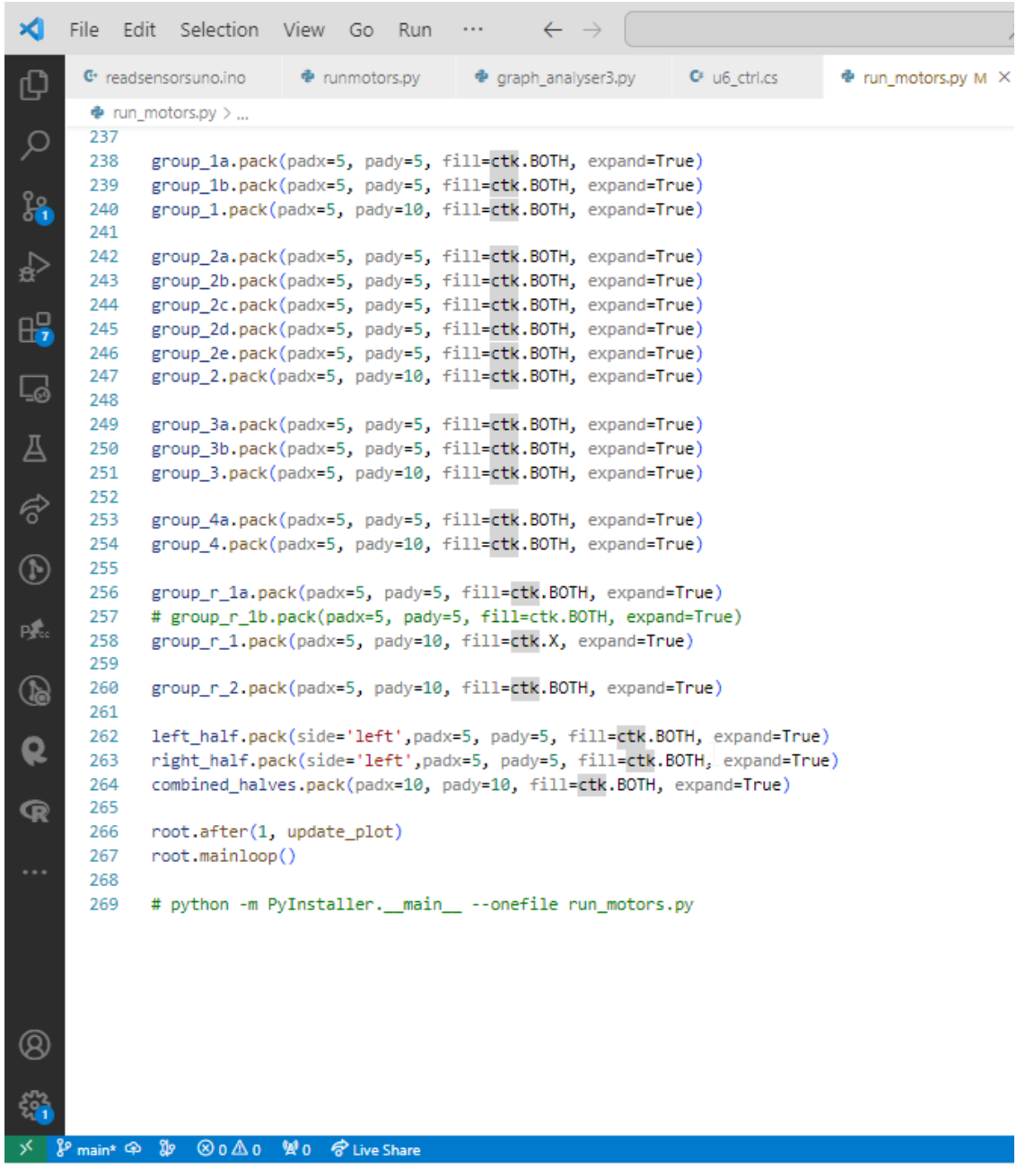

File Edit Selection View Go Run
graph_analyser3.py
u6_ctrl.cs
run_motors.py M
u6_but
run_motors.py > ...
42
43 def update_Theme():
44 theme = "dark" if dark_mode else "light"
45 root.configure(background='black' if dark_mode else 'light')
46 ctk.set_appearance_mode(theme)
47 print(f"SWitching to {theme} Mode")
48 switch_1.configure(text='Light Mode' if dark_mode else 'Dark Mode')
49
50 # update the canvas themes
51 canvas3.get_tk_widget().configure(bg = 'black' if dark_mode else 'white')
52 def start_plotting():
53 global is_plotting
54 is_plotting = True
55 thread = threading.Thread(target=update_plot)
56 thread.daemon = True
57 thread.start()
58 ser.reset_input_buffer()
59
60 # Function to stop reading data from Arduino
61 def stop_plotting():
62 global is_plotting
63 is_plotting = False
64
65 # Function to update the first plot
66 def update_plot():
67 global is_plotting,pulses,csv_data, data_counter
68 while is_plotting:
69 try:
70 data = ser.readline().decode('utf-8').strip()
71 comb_data= data.split(',')
72 #if(len(comb_data)==3):
73
74 pwm = comb_data[0]
75
76 with data_lock:
77 if len(pulses) < 50:
78 pulses = np.append(pulses, int(pwm[0:4]))
79 else:
80 pulses[0:49] = pulses[1:50]
81 pulses[49] = float(pwm[0:4])
82
83 # pwm_label.ure(text=f'PWM: {pwm}')
main*
0 0
0
Live Share


File Edit Selection View Go Run
graph_analyser3.py
u6_ctrl.cs
run_motors.py M
run_motors.py > ...
pwm_label.configure(text=f'PWM: {pwm}')
# Append data to csv_data
csv_data.append([pwm])
# Check if data exceeds 200 points and save to a new CSV file
if len(csv_data) >= max_data_points:
save_data_to_csv(data_counter)
data_counter += 1
csv_data = []
root.after(1, update_plot3)
except Exception as e:
print(e)
#root.update()
# Function to save data to a CSV file with a unique identifier
def save_data_to_csv(counter):
filename = f'datapwm{counter}.csv'
with open(filename, 'w', newline='') as csvfile:
csvwriter = csv.writer(csvfile)
csvwriter.writerow(['PWM'])
for data_point in csv_data:
csvwriter.writerow(data_point)
# Start a separate thread to continuously update data and plots
data_thread = threading.Thread(target=update_plot)
data_thread.daemon = True
data_thread.start()
# Function to update the third plot
def update_plot3():
with data_lock:
lines3.set_xdata(np.arange(0, len(pulses)))
lines3.set_ydata(pulses)
canvas3.draw()
# Function to send commands to Arduino
def send_command(command):
ser.write(command.encode('utf-8'))
# Function to update the serial connection with selected COM port and baud rate
main*
Live Share

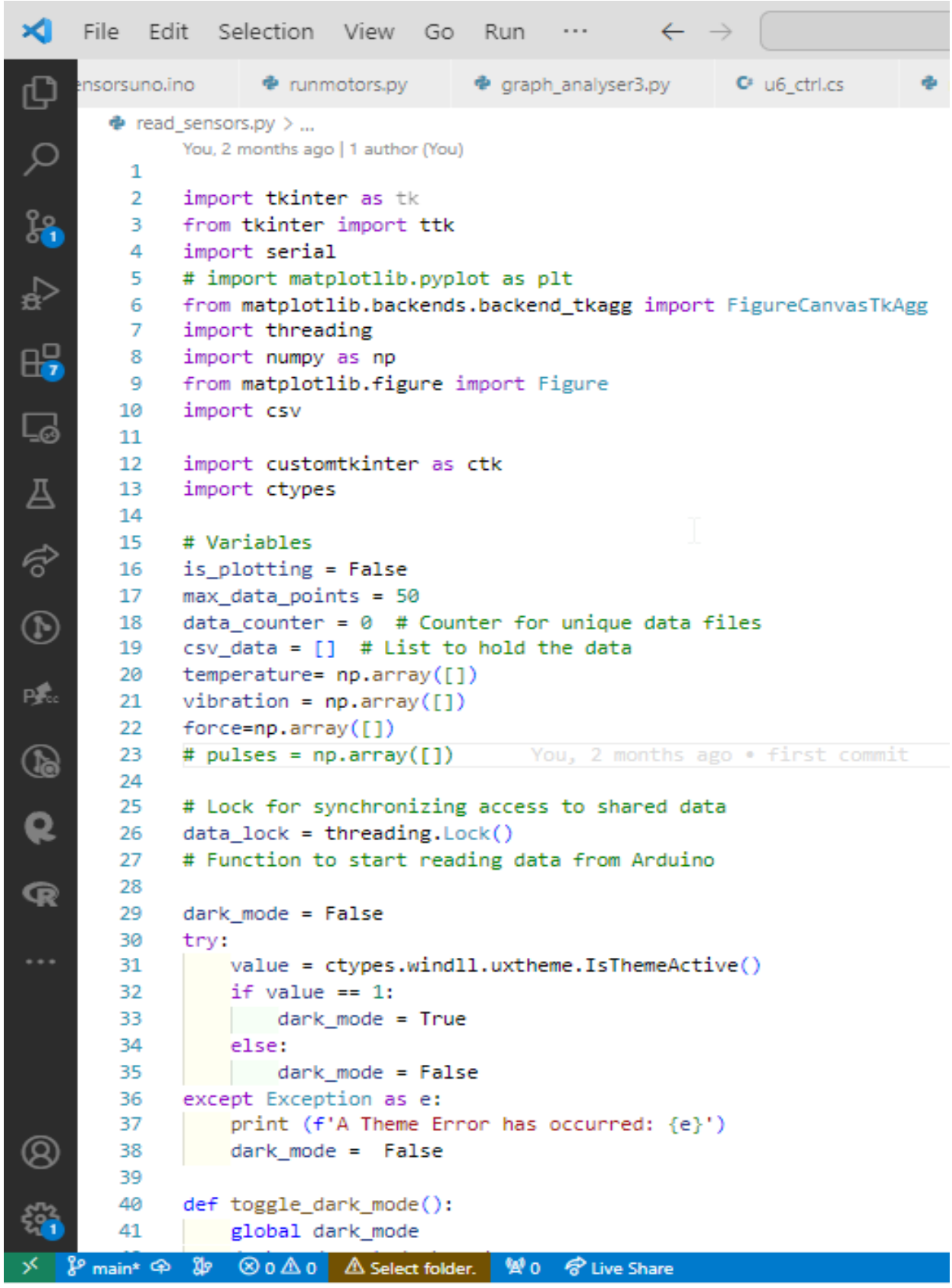

File Edit Selection View Go Run ···
graph_analyser3.py
u6_ctrl.cs
run_motors.py M
u6_buti
run_motors.py > ...
124
125 # Function to update the serial connection with selected COM port and baud rate
126 def update_serial_connection():
127 selected_com_port = com_port_combo.get()
128 selected_baud_rate = baud_rate_combo.get()
129
130 global ser
131 try:
132 ser.close()
133 ser = serial.Serial(selected_com_port, int(selected_baud_rate))
134 ser.reset_input_buffer()
135 print(f'Connection on {ser} successful')
136 except serial.SerialException as e:
137 print(f'Error: {e}. Connection on {selected_com_port} not found.')
138 except Exception as e:
139 print(f'An unexpected error occurred: {e}')
140
141 # Initialize tkinter window
142 root = ctk.CTk()
143 root.title("MOTOR CONTROL GUI: ONE AXIS AUTOMATED DRILL")
144 root.configure(background='lightblue')
145
146 # Serial communication setup
147 ser = serial.Serial() # Initialize with default values
148 #ser.reset_input_buffer()
149
150 # Create GUI components
151 root.update()
152
153 combined_halves = ctk.CTkFrame(root)
154 left_half = ctk.CTkFrame(combined_halves)
155 right_half = ctk.CTkFrame(combined_halves)
156
157 group_1 = ctk.CTkFrame(left_half)
158 group_1a = ctk.CTkFrame(group_1)
159 group_1b = ctk.CTkFrame(group_1)
160
161 group_2=ctk.CTkFrame(left_half)
162 group_2a=ctk.CTkFrame(group_2)
163 group_2b=ctk.CTkFrame(group_2)
164 group_2c=ctk.CTkFrame(group_2)
165 group_2d=ctk.CTkFrame(group_2)
main*
0 0
0
Live Share

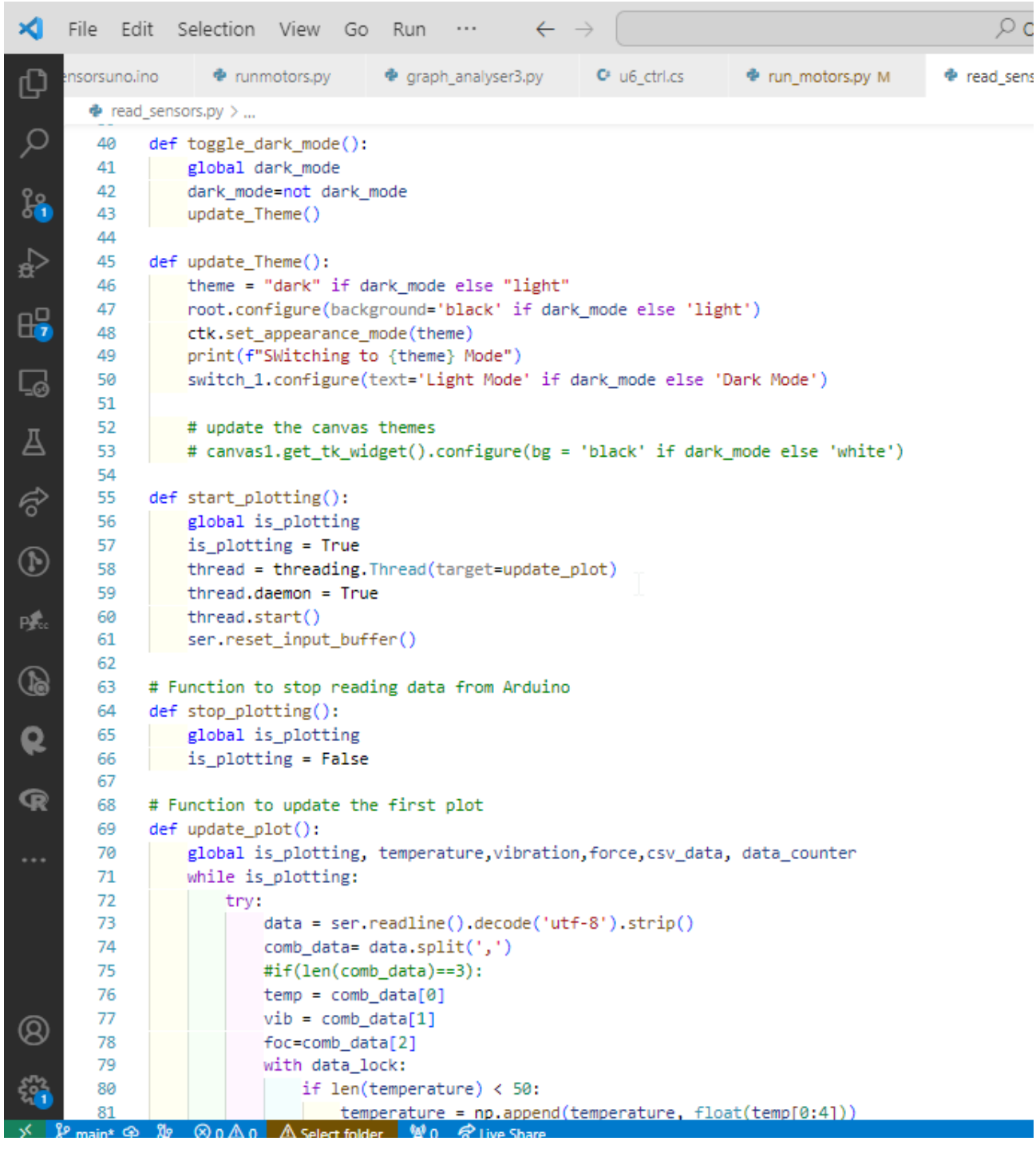

File Edit Selection View Go Run ···
One_Axis_Final
readsensorsuno.ino
runmotors.py
graph_analyser3.py
u6_ctrl.cs
run_motors.py M
u6_button_ctrl.cs
run_motors.py > ...
group_2b=ctk.CTkFrame(group_2)
group_2c=ctk.CTkFrame(group_2)
group_2d=ctk.CTkFrame(group_2)
group_2e=ctk.CTkFrame(group_2)
group_3=ctk.CTkFrame(left_half)
group_3a=ctk.CTkFrame(group_3)
group_3b=ctk.CTkFrame(group_3)
group_4=ctk.CTkFrame(left_half)
group_4a=ctk.CTkFrame(group_4)
group_r_1=ctk.CTkFrame(right_half)
group_r_1a=ctk.CTkFrame(group_r_1)
group_r_1b=ctk.CTkFrame(group_r_1)
group_r_2=ctk.CTkFrame(right_half)
start_button = ctk.CTkButton(group_r_1a, text="Start Plot", command=start_plotting)
stop_button = ctk.CTkButton(group_r_1a, text="Stop Plot", command=stop_plotting)
pwm_label = ctk.CTkLabel(group_4a, text="PWM: ",font=("Helvetica",15))
(function) def send_command
fast_drill_speed_button = ctk.CTkButton(group_3a, text="FAST DRILL SPEED", command=lambda:
slow_drill_speed_button = ctk.CTkButton(group_3b, text="SLOW DRILL SPEED", command=lambda: send_command('slowdrill'))
stop_machine_button = ctk.CTkButton(group_1a, text="STOP MACHINE", command=lambda: send_command('stopmachine'))
switch_1 = ctk.CTkSwitch(master=group_1b, text='Light Mode' if dark_mode else 'Dark Mode' , command=toggle_dark_mode)
# Dropdown menu for COM port selection
com_port_label = ctk.CTkLabel(group_2a, text="Select COM Port:")
com_port_label.grid(row=1, column=0, sticky="w",padx=10,pady=5)
com_ports = ["COM1", "COM2", "COM3", "COM4","COM5","COM6",
"COM7","COM8","COM9","COM10","COM11","COM12","COM13",
"COM14","COM15","COM16","COM17","COM18","COM19","COM20"]
com_port_combo = ttk.Combobox(group_2b, values=com_ports)
com_port_combo.set("COM1") # Set a default COM port
# Dropdown menu for baud rate selection
baud_rate_label = ctk.CTkLabel(group_2c, text="Select Baud Rate:")
baud_rate_label.grid(row=2, column=0, sticky="w",padx=10,pady=5)
baud_rates = ["9600", "115200", "57600", "38400"] # Replace with your available baud rates
baud_rate_combo = ttk.Combobox(group_2d, values=baud_rates)
baud_rate_combo.set("9600") # Set a default baud rate
main*
0 0
0
Live Share
You,

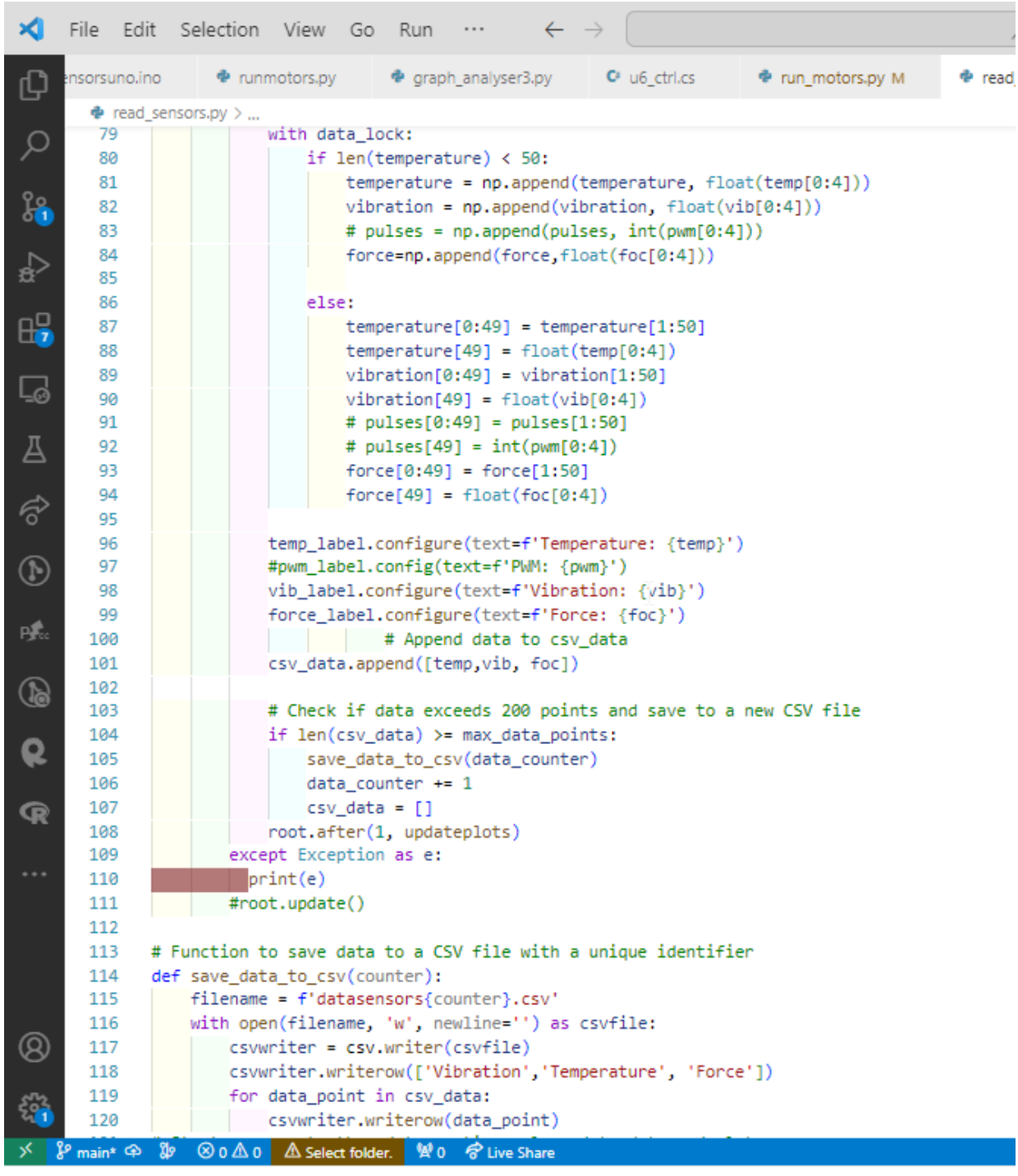

File Edit Selection View Go Run ···
One_Axis_Final
readsensorsuno.ino
runmotors.py
graph_analyser3.py
u6_ctrl.cs
run_motors.py M
u6_button_ctrl.cs
run_motors.py > ...
baud_rate_combo.set("9600") # Set a default baud rate
# Button to update the serial connection with selected COM port and baud rate
update_button = ctk.CTkButton(group_2e, text="Update Serial Connection", command=update_serial_connection)
# Matplotlib setup for the third plot
fig3 = Figure(figsize=(6,4))
ax3 = fig3.add_subplot(111)
ax3.set_title("PWM")
#ax3.set_xlabel("Time")
ax3.set_ylabel("pwm")
ax3.grid(True)
ax3.set_xlim([0, 50])
ax3.set_ylim([0, 260])
lines3 = ax3.plot([], [])[0]
canvas3 = FigureCanvasTkAgg(fig3, master=group_r_2)
canvas3.draw()
# Grid layout
com_port_label.pack(padx=5, pady=5)
com_port_combo.pack(padx=5,pady=5)
baud_rate_label.pack(padx=5,pady=5)
switch_1.pack(padx=10, pady=10)
start_button.pack(side='left',padx=5, pady=5)
stop_button.pack(side='right',padx=5, pady=5)
pwm_label.pack(padx=5, pady=5)
baud_rate_combo.pack(padx=5,pady=5)
update_button.pack(padx=5, pady=5)
fast_drill_speed_button.pack(padx=10, pady=10)
slow_drill_speed_button.pack(padx=10, pady=10)
stop_machine_button.pack(padx=10, pady=10)
canvas3.get_tk_widget().pack(padx=10, pady=10)
group_1a.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
group_1b.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
group_1.pack(padx=5, pady=10, fill=ctk.BOTH, expand=True)
group_2a.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
group_2b.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
group_2c.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
main*
0 0
0
Live Share

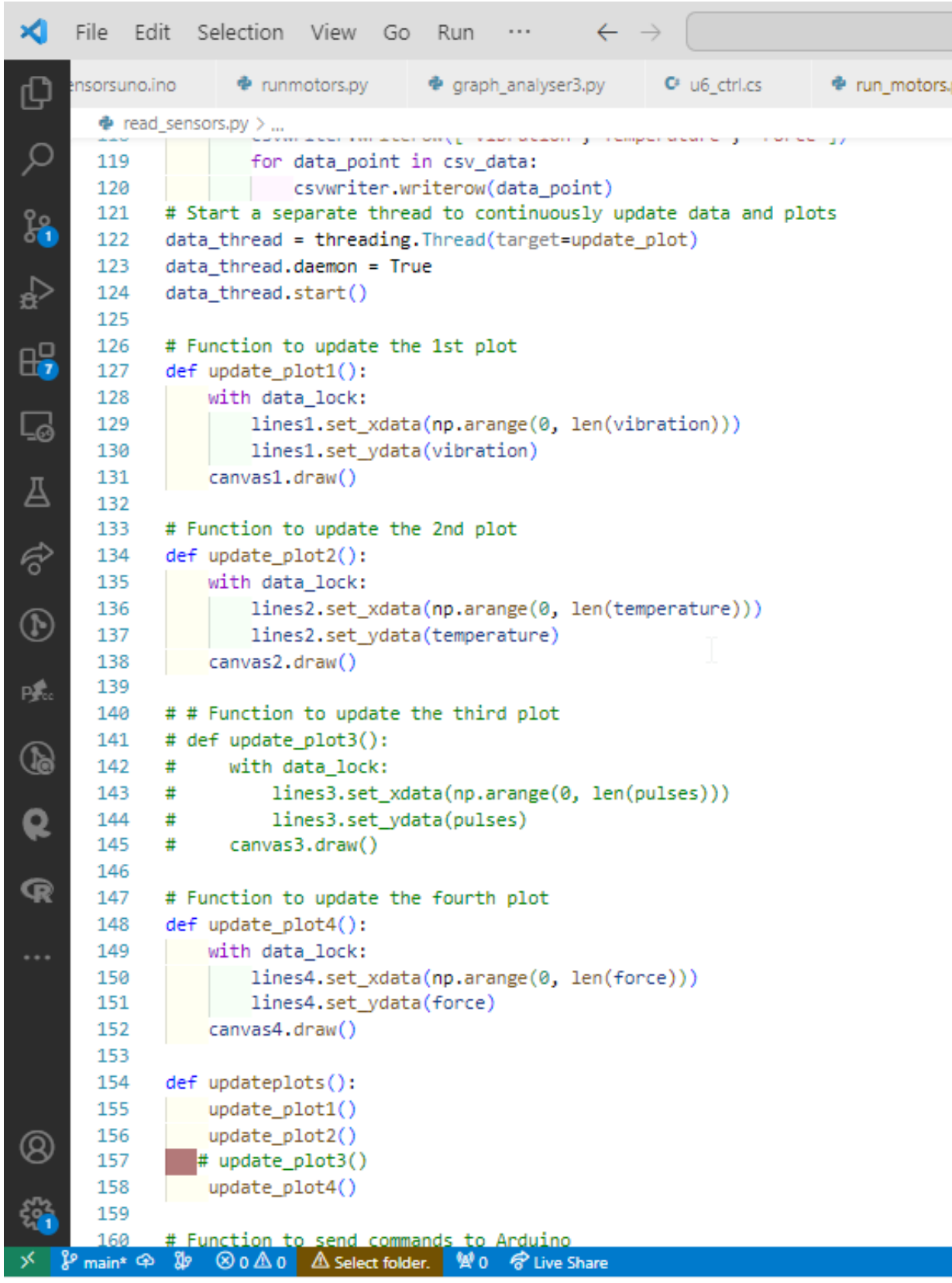

File Edit Selection View Go Run ···
readsensorsuno.ino
runmotors.py
graph_analyser3.py
u6_ctrl.cs
run_motors.py M
run_motors.py > ...
237
238 group_1a.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
239 group_1b.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
240 group_1.pack(padx=5, pady=10, fill=ctk.BOTH, expand=True)
241
242 group_2a.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
243 group_2b.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
244 group_2c.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
245 group_2d.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
246 group_2e.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
247 group_2.pack(padx=5, pady=10, fill=ctk.BOTH, expand=True)
248
249 group_3a.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
250 group_3b.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
251 group_3.pack(padx=5, pady=10, fill=ctk.BOTH, expand=True)
252
253 group_4a.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
254 group_4.pack(padx=5, pady=10, fill=ctk.BOTH, expand=True)
255
256 group_r_1a.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
257 # group_r_1b.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
258 group_r_1.pack(padx=5, pady=10, fill=ctk.X, expand=True)
259
260 group_r_2.pack(padx=5, pady=10, fill=ctk.BOTH, expand=True)
261
262 left_half.pack(side='left',padx=5, pady=5, fill=ctk.BOTH, expand=True)
263 right_half.pack(side='left',padx=5, pady=5, fill=ctk.BOTH, expand=True)
264 combined_halves.pack(padx=10, pady=10, fill=ctk.BOTH, expand=True)
265
266 root.after(1, update_plot)
267 root.mainloop()
268
269 # python -m PyInstaller.__main__ --onefile run_motors.py
main*
0 0
0
Live Share

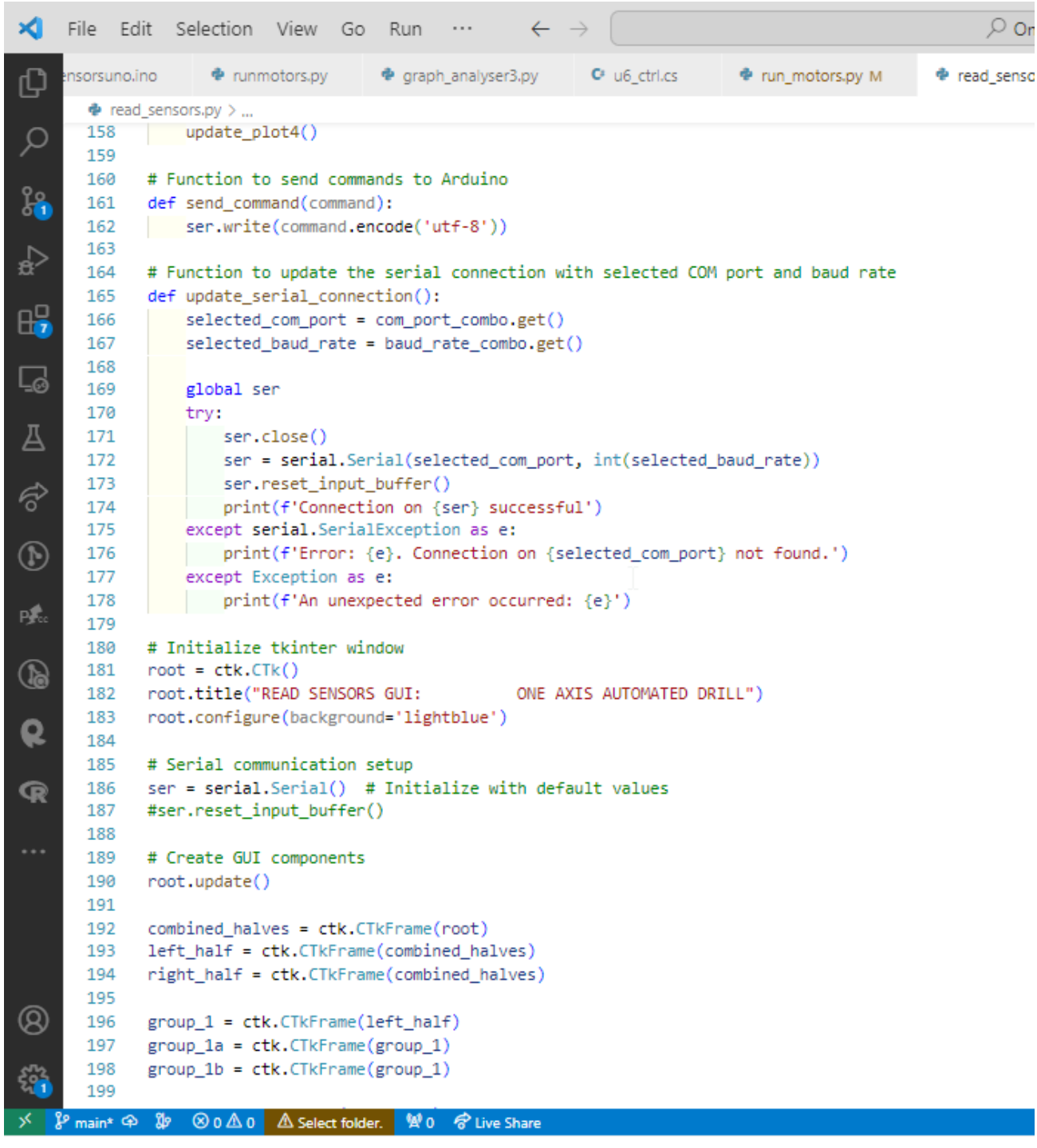
2) The code used to read sensors and to provide the read sensors GUI

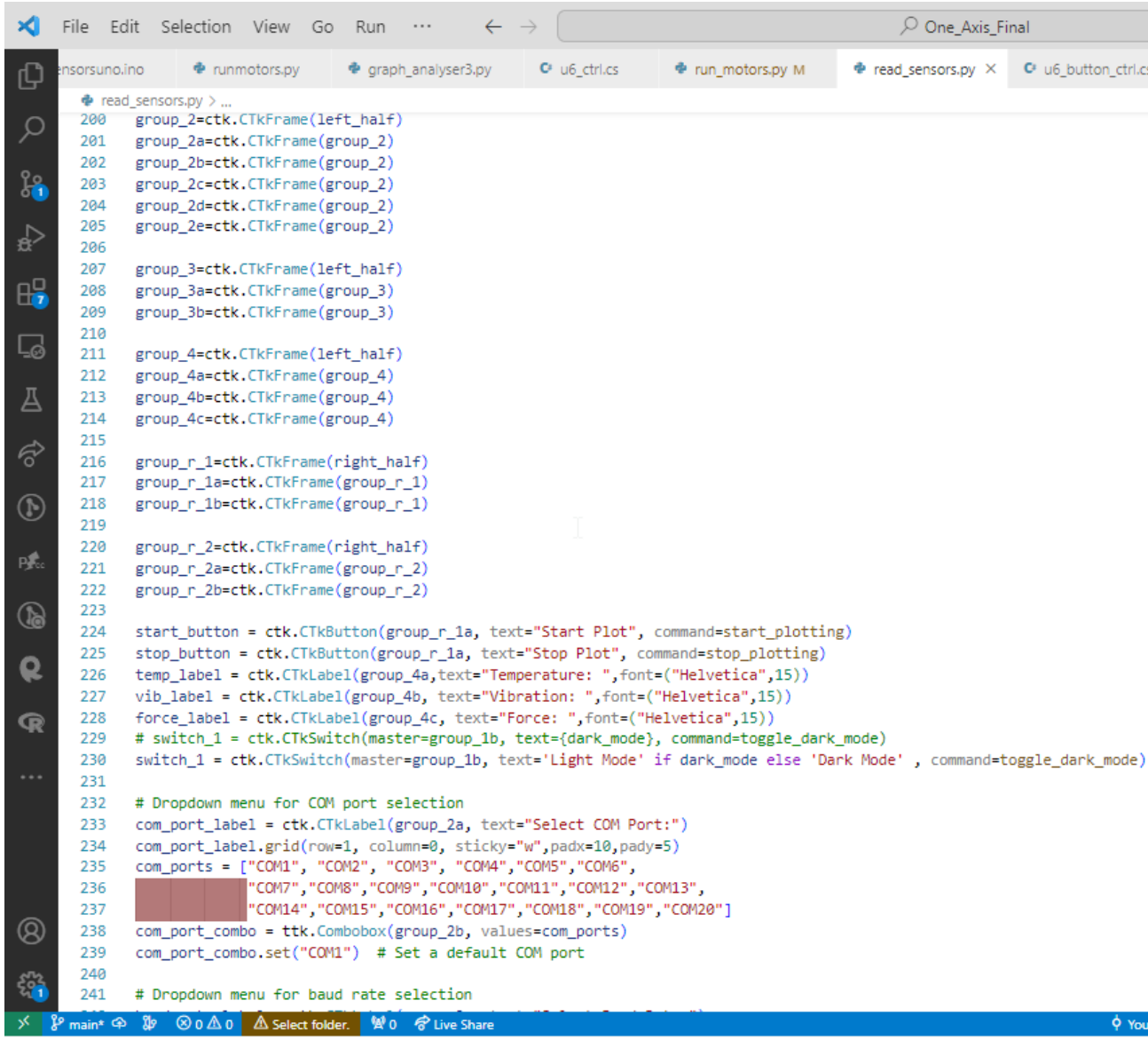


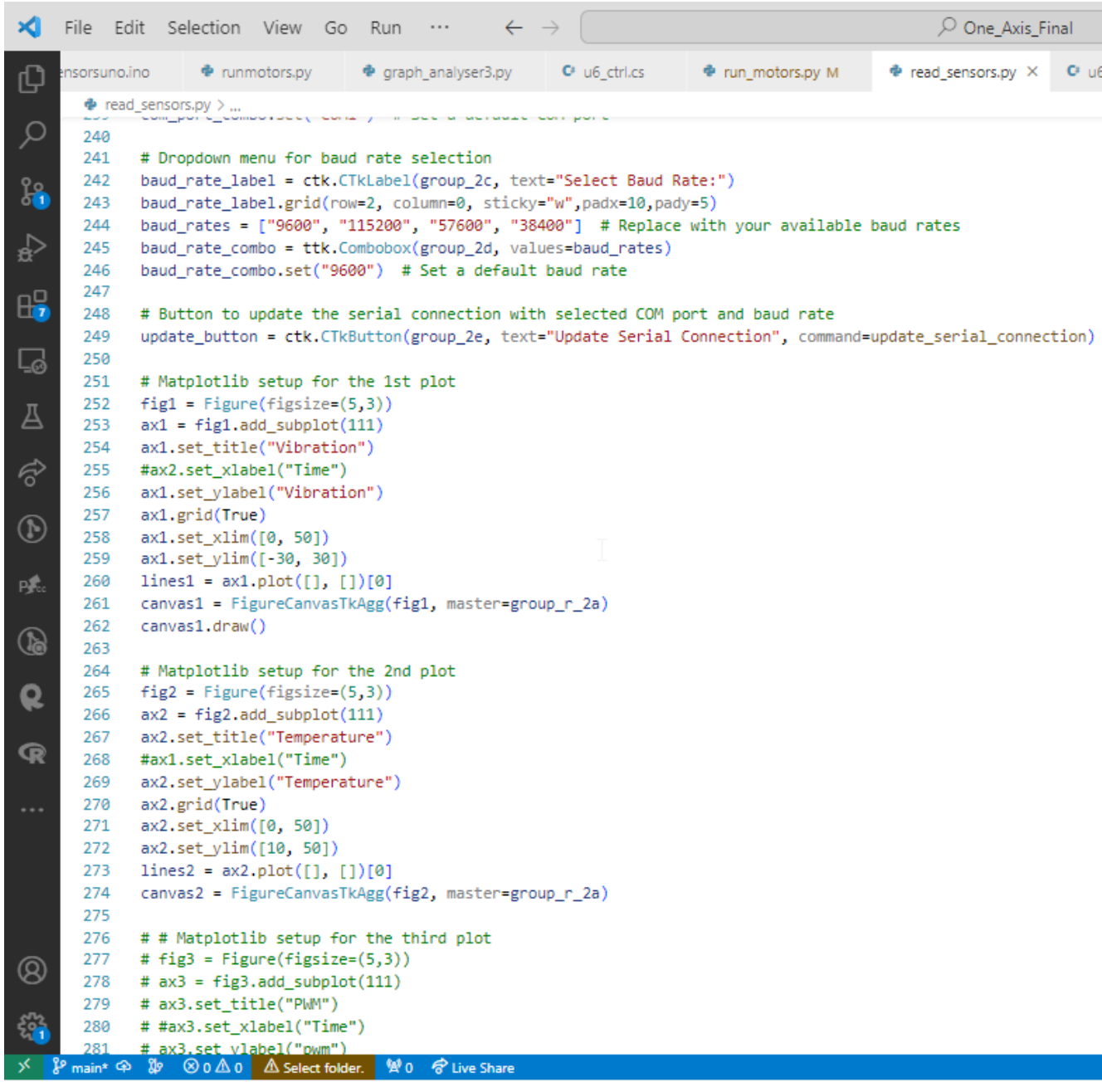

File Edit Selection View Go Run
nsorsuno.ino
runmotors.py
graph_analyser3.py
u6_ctrl.cs
run_motors.py M
read_sens
read_sensors.py > ...
40 def toggle_dark_mode():
41 global dark_mode
42 dark_mode=not dark_mode
43 update_Theme()
44
45 def update_Theme():
46 theme = "dark" if dark_mode else "light"
47 root.configure(background='black' if dark_mode else 'light')
48 ctk.set_appearance_mode(theme)
49 print(f"SWitching to {theme} Mode")
50 switch_1.configure(text='Light Mode' if dark_mode else 'Dark Mode')
51
52 # update the canvas themes
53 # canvas1.get_tk_widget().configure(bg = 'black' if dark_mode else 'white')
54
55 def start_plotting():
56 global is_plotting
57 is_plotting = True
58 thread = threading.Thread(target=update_plot)
59 thread.daemon = True
60 thread.start()
61 ser.reset_input_buffer()
62
63 # Function to stop reading data from Arduino
64 def stop_plotting():
65 global is_plotting
66 is_plotting = False
67
68 # Function to update the first plot
69 def update_plot():
70 global is_plotting, temperature,vibration,force,csv_data, data_counter
71 while is_plotting:
72 try:
73 data = ser.readline().decode('utf-8').strip()
74 comb_data= data.split(',')
75 #if(len(comb_data)==3):
76 temp = comb_data[0]
77 vib = comb_data[1]
78 foc=comb_data[2]
79 with data_lock:
80 if len(temperature) < 50:
81 temperature = np.append(temperature, float(temp[0:4]))
main*
Select folder
Live Share

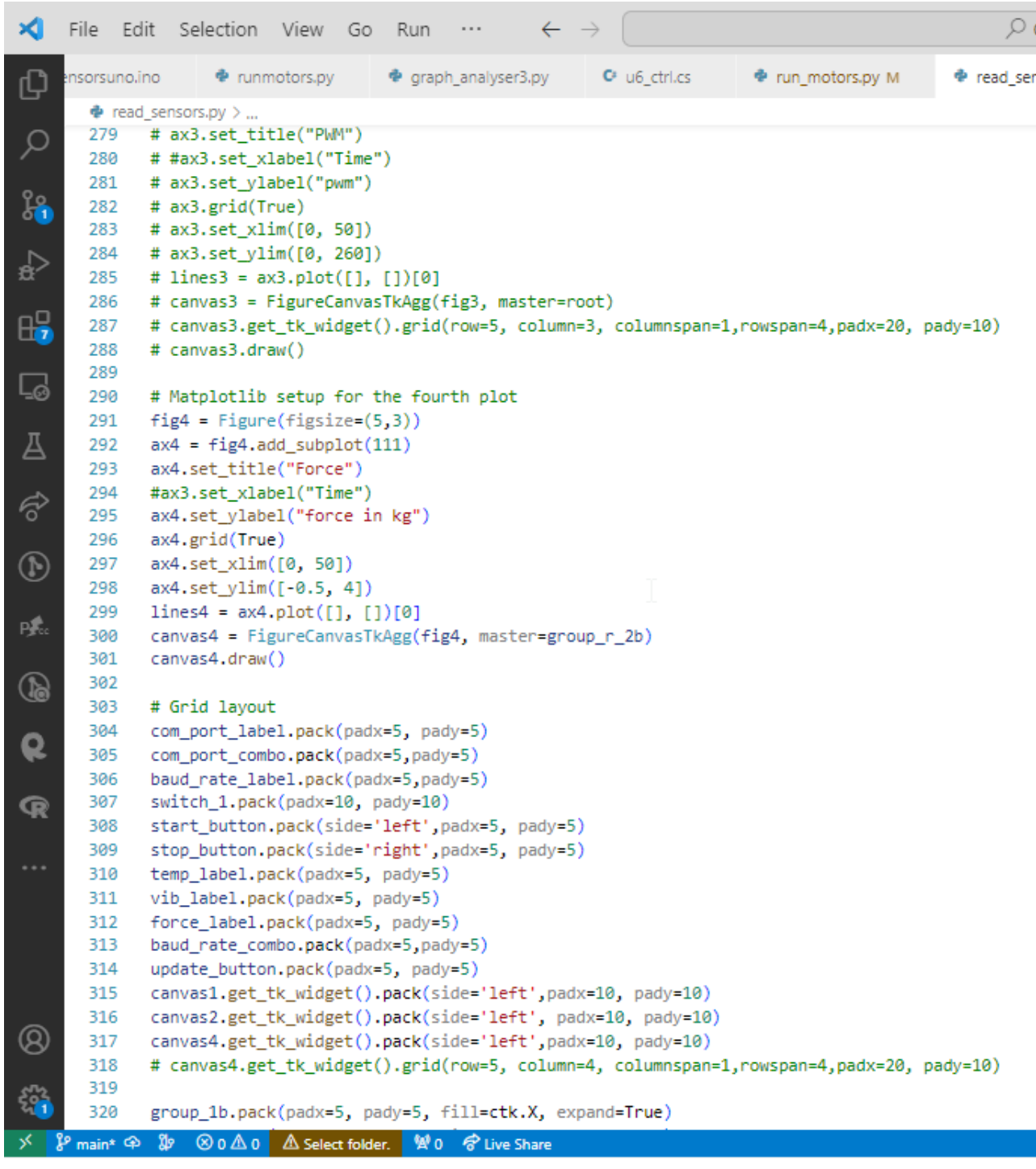

File Edit Selection View Go Run ···
nsorsuno.ino
runmotors.py
graph_analyser3.py
u6_ctrl.cs
run_motors.py M
read
read_sensors.py > ...
79 with data_lock:
80 if len(temperature) < 50:
81 temperature = np.append(temperature, float(temp[0:4]))
82 vibration = np.append(vibration, float(vib[0:4]))
83 # pulses = np.append(pulses, int(pwm[0:4]))
84 force=np.append(force,float(foc[0:4]))
85
86 else:
87 temperature[0:49] = temperature[1:50]
88 temperature[49] = float(temp[0:4])
89 vibration[0:49] = vibration[1:50]
90 vibration[49] = float(vib[0:4])
91 # pulses[0:49] = pulses[1:50]
92 # pulses[49] = int(pwm[0:4])
93 force[0:49] = force[1:50]
94 force[49] = float(foc[0:4])
95
96 temp_label.configure(text=f'Temperature: {temp}')
97 #pwm_label.config(text=f'PWM: {pwm}')
98 vib_label.configure(text=f'Vibration: {vib}')
99 force_label.configure(text=f'Force: {foc}')
100 # Append data to csv_data
101 csv_data.append([temp,vib, foc])
102
103 # Check if data exceeds 200 points and save to a new CSV file
104 if len(csv_data) >= max_data_points:
105 save_data_to_csv(data_counter)
106 data_counter += 1
107 csv_data = []
108 root.after(1, updateplots)
109 except Exception as e:
110 print(e)
111 #root.update()
112
113 # Function to save data to a CSV file with a unique identifier
114 def save_data_to_csv(counter):
115 filename = f'datasensors{counter}.csv'
116 with open(filename, 'w', newline='') as csvfile:
117 csvwriter = csv.writer(csvfile)
118 csvwriter.writerow(['Vibration','Temperature', 'Force'])
119 for data_point in csv_data:
120 csvwriter.writerow(data_point)
main*
0 0
Select folder.
0
Live Share

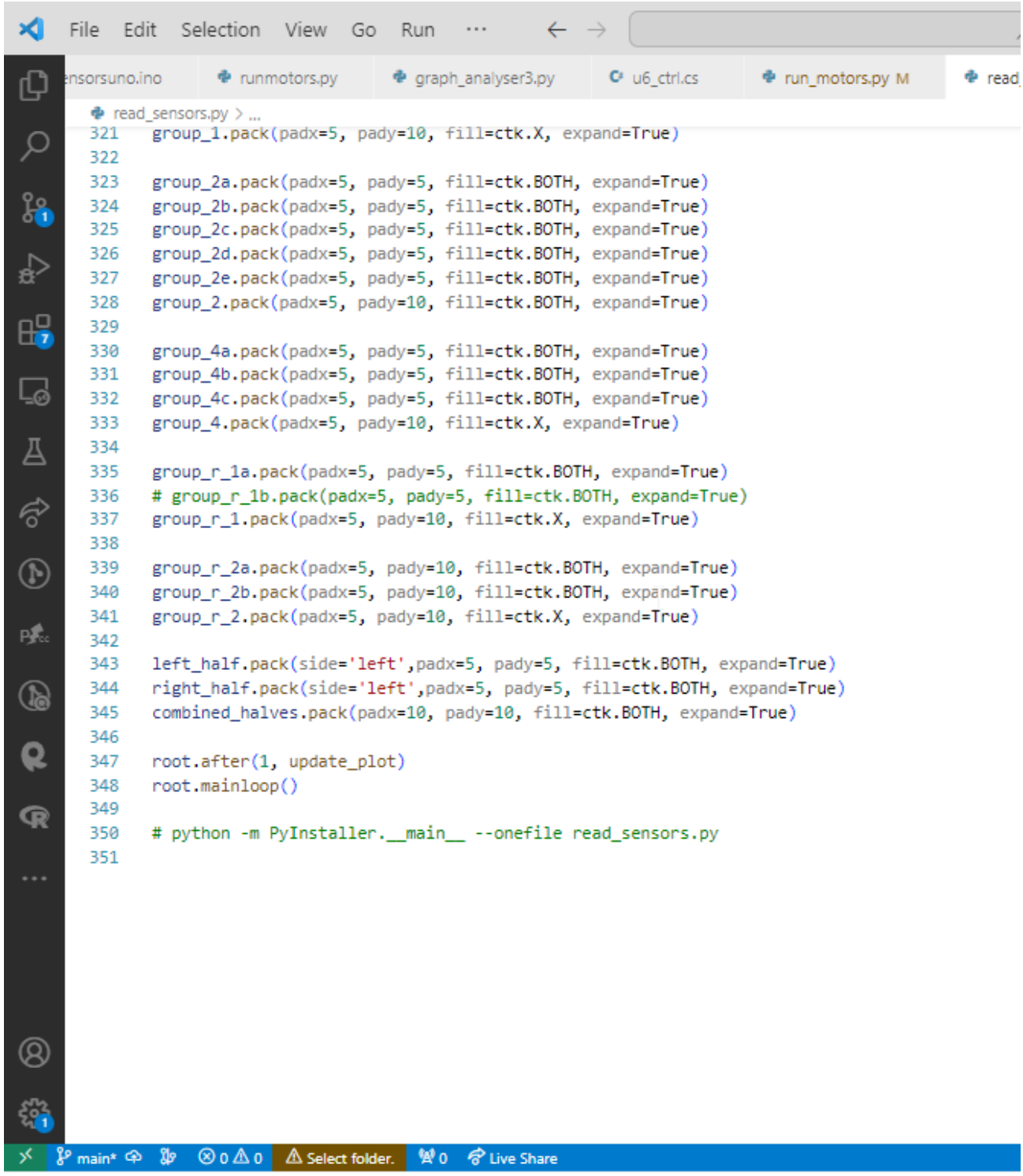

File Edit Selection View Go Run ···
nsorsuno.ino
runmotors.py
graph_analyser3.py
u6_ctrl.cs
run_motors.p
read_sensors.py > ...
for data_point in csv_data:
csvwriter.writerow(data_point)
# Start a separate thread to continuously update data and plots
data_thread = threading.Thread(target=update_plot)
data_thread.daemon = True
data_thread.start()
# Function to update the 1st plot
def update_plot1():
with data_lock:
lines1.set_xdata(np.arange(0, len(vibration)))
lines1.set_ydata(vibration)
canvas1.draw()
# Function to update the 2nd plot
def update_plot2():
with data_lock:
lines2.set_xdata(np.arange(0, len(temperature)))
lines2.set_ydata(temperature)
canvas2.draw()
# # Function to update the third plot
# def update_plot3():
# with data_lock:
# lines3.set_xdata(np.arange(0, len(pulses)))
# lines3.set_ydata(pulses)
# canvas3.draw()
# Function to update the fourth plot
def update_plot4():
with data_lock:
lines4.set_xdata(np.arange(0, len(force)))
lines4.set_ydata(force)
canvas4.draw()
def updateplots():
update_plot1()
update_plot2()
# update_plot3()
update_plot4()
# Function to send commands to Arduino
main*
0 0
Select folder.
0
Live Share


File Edit Selection View Go Run
nsorsuno.ino
runmotors.py
graph_analyser3.py
u6_ctrl.cs
run_motors.py M
read_senso
read_sensors.py > ...
158 update_plot4()
159
160 # Function to send commands to Arduino
161 def send_command(command):
162 ser.write(command.encode('utf-8'))
163
164 # Function to update the serial connection with selected COM port and baud rate
165 def update_serial_connection():
166 selected_com_port = com_port_combo.get()
167 selected_baud_rate = baud_rate_combo.get()
168
169 global ser
170 try:
171 ser.close()
172 ser = serial.Serial(selected_com_port, int(selected_baud_rate))
173 ser.reset_input_buffer()
174 print(f'Connection on {ser} successful')
175 except serial.SerialException as e:
176 print(f'Error: {e}. Connection on {selected_com_port} not found.')
177 except Exception as e:
178 print(f'An unexpected error occurred: {e}')
179
180 # Initialize tkinter window
181 root = ctk.CTk()
182 root.title("READ SENSORS GUI: ONE AXIS AUTOMATED DRILL")
183 root.configure(background='lightblue')
184
185 # Serial communication setup
186 ser = serial.Serial() # Initialize with default values
187 #ser.reset_input_buffer()
188
189 # Create GUI components
190 root.update()
191
192 combined_halves = ctk.CTkFrame(root)
193 left_half = ctk.CTkFrame(combined_halves)
194 right_half = ctk.CTkFrame(combined_halves)
195
196 group_1 = ctk.CTkFrame(left_half)
197 group_1a = ctk.CTkFrame(group_1)
198 group_1b = ctk.CTkFrame(group_1)
199
main* 0 0 Select folder. 0 Live Share


File Edit Selection View Go Run ···
One_Axis_Final
nsorsuno.ino
runmotors.py
graph_analyser3.py
u6_ctrl.cs
run_motors.py M
read_sensors.py
u6_button_ctrl.cs
read_sensors.py > ...
200 group_2=ctk.CTkFrame(left_half)
201 group_2a=ctk.CTkFrame(group_2)
202 group_2b=ctk.CTkFrame(group_2)
203 group_2c=ctk.CTkFrame(group_2)
204 group_2d=ctk.CTkFrame(group_2)
205 group_2e=ctk.CTkFrame(group_2)
206
207 group_3=ctk.CTkFrame(left_half)
208 group_3a=ctk.CTkFrame(group_3)
209 group_3b=ctk.CTkFrame(group_3)
210
211 group_4=ctk.CTkFrame(left_half)
212 group_4a=ctk.CTkFrame(group_4)
213 group_4b=ctk.CTkFrame(group_4)
214 group_4c=ctk.CTkFrame(group_4)
215
216 group_r_1=ctk.CTkFrame(right_half)
217 group_r_1a=ctk.CTkFrame(group_r_1)
218 group_r_1b=ctk.CTkFrame(group_r_1)
219
220 group_r_2=ctk.CTkFrame(right_half)
221 group_r_2a=ctk.CTkFrame(group_r_2)
222 group_r_2b=ctk.CTkFrame(group_r_2)
223
224 start_button = ctk.CTkButton(group_r_1a, text="Start Plot", command=start_plotting)
225 stop_button = ctk.CTkButton(group_r_1a, text="Stop Plot", command=stop_plotting)
226 temp_label = ctk.CTkLabel(group_4a,text="Temperature: ",font=("Helvetica",15))
227 vib_label = ctk.CTkLabel(group_4b, text="Vibration: ",font=("Helvetica",15))
228 force_label = ctk.CTkLabel(group_4c, text="Force: ",font=("Helvetica",15))
229 # switch_1 = ctk.CTkSwitch(master=group_1b, text={dark_mode}, command=toggle_dark_mode)
230 switch_1 = ctk.CTkSwitch(master=group_1b, text='Light Mode' if dark_mode else 'Dark Mode' , command=toggle_dark_mode)
231
232 # Dropdown menu for COM port selection
233 com_port_label = ctk.CTkLabel(group_2a, text="Select COM Port:")
234 com_port_label.grid(row=1, column=0, sticky="w",padx=10,pady=5)
235 com_ports = ["COM1", "COM2", "COM3", "COM4","COM5","COM6",
236 "COM7","COM8","COM9","COM10","COM11","COM12","COM13",
237 "COM14","COM15","COM16","COM17","COM18","COM19","COM20"]
238 com_port_combo = ttk.Combobox(group_2b, values=com_ports)
239 com_port_combo.set("COM1") # Set a default COM port
240
241 # Dropdown menu for baud rate selection
main* 0 0 Select folder. 0 Live Share You,


File Edit Selection View Go Run ···
One_Axis_Final
ensorsuno.ino
runmotors.py
graph_analyser3.py
u6_ctrl.cs
run_motors.py M
read_sensors.py
read_sensors.py > ...
240
241 # Dropdown menu for baud rate selection
242 baud_rate_label = ctk.CTkLabel(group_2c, text="Select Baud Rate:")
243 baud_rate_label.grid(row=2, column=0, sticky="w",padx=10,pady=5)
244 baud_rates = ["9600", "115200", "57600", "38400"] # Replace with your available baud rates
245 baud_rate_combo = ttk.Combobox(group_2d, values=baud_rates)
246 baud_rate_combo.set("9600") # Set a default baud rate
247
248 # Button to update the serial connection with selected COM port and baud rate
249 update_button = ctk.CTkButton(group_2e, text="Update Serial Connection", command=update_serial_connection)
250
251 # Matplotlib setup for the 1st plot
252 fig1 = Figure(figsize=(5,3))
253 ax1 = fig1.add_subplot(111)
254 ax1.set_title("Vibration")
255 #ax2.set_xlabel("Time")
256 ax1.set_ylabel("Vibration")
257 ax1.grid(True)
258 ax1.set_xlim([0, 50])
259 ax1.set_ylim([-30, 30])
260 lines1 = ax1.plot([], [])[0]
261 canvas1 = FigureCanvasTkAgg(fig1, master=group_r_2a)
262 canvas1.draw()
263
264 # Matplotlib setup for the 2nd plot
265 fig2 = Figure(figsize=(5,3))
266 ax2 = fig2.add_subplot(111)
267 ax2.set_title("Temperature")
268 #ax1.set_xlabel("Time")
269 ax2.set_ylabel("Temperature")
270 ax2.grid(True)
271 ax2.set_xlim([0, 50])
272 ax2.set_ylim([10, 50])
273 lines2 = ax2.plot([], [])[0]
274 canvas2 = FigureCanvasTkAgg(fig2, master=group_r_2a)
275
276 # # Matplotlib setup for the third plot
277 # fig3 = Figure(figsize=(5,3))
278 # ax3 = fig3.add_subplot(111)
279 # ax3.set_title("PWM")
280 # #ax3.set_xlabel("Time")
281 # ax3.set_ylabel("pwm")
main*
0 0
Select folder.
0
Live Share


File Edit Selection View Go Run
nsorsuno.ino
runmotors.py
graph_analyser3.py
u6_ctrl.cs
run_motors.py M
read_sens
read_sensors.py > ...
# ax3.set_title("PWM")
# #ax3.set_xlabel("Time")
# ax3.set_ylabel("pwm")
# ax3.grid(True)
# ax3.set_xlim([0, 50])
# ax3.set_ylim([0, 260])
# lines3 = ax3.plot([], [])[0]
# canvas3 = FigureCanvasTkAgg(fig3, master=root)
# canvas3.get_tk_widget().grid(row=5, column=3, columnspan=1,rowspan=4,padx=20, pady=10)
# canvas3.draw()
# Matplotlib setup for the fourth plot
fig4 = Figure(figsize=(5,3))
ax4 = fig4.add_subplot(111)
ax4.set_title("Force")
#ax3.set_xlabel("Time")
ax4.set_ylabel("force in kg")
ax4.grid(True)
ax4.set_xlim([0, 50])
ax4.set_ylim([-0.5, 4])
lines4 = ax4.plot([], [])[0]
canvas4 = FigureCanvasTkAgg(fig4, master=group_r_2b)
canvas4.draw()
# Grid layout
com_port_label.pack(padx=5, pady=5)
com_port_combo.pack(padx=5,pady=5)
baud_rate_label.pack(padx=5,pady=5)
switch_1.pack(padx=10, pady=10)
start_button.pack(side='left',padx=5, pady=5)
stop_button.pack(side='right',padx=5, pady=5)
temp_label.pack(padx=5, pady=5)
vib_label.pack(padx=5, pady=5)
force_label.pack(padx=5, pady=5)
baud_rate_combo.pack(padx=5,pady=5)
update_button.pack(padx=5, pady=5)
canvas1.get_tk_widget().pack(side='left',padx=10, pady=10)
canvas2.get_tk_widget().pack(side='left', padx=10, pady=10)
canvas4.get_tk_widget().pack(side='left',padx=10, pady=10)
# canvas4.get_tk_widget().grid(row=5, column=4, columnspan=1,rowspan=4,padx=20, pady=10)
group_1b.pack(padx=5, pady=5, fill=ctk.X, expand=True)
main*
0 0
Select folder.
0
Live Share


File Edit Selection View Go Run ···
nsorsuno.ino
runmotors.py
graph_analyser3.py
u6_ctrl.cs
run_motors.py M
read
read_sensors.py > ...
group_1.pack(padx=5, pady=10, fill=ctk.X, expand=True)
group_2a.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
group_2b.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
group_2c.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
group_2d.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
group_2e.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
group_2.pack(padx=5, pady=10, fill=ctk.BOTH, expand=True)
group_4a.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
group_4b.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
group_4c.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
group_4.pack(padx=5, pady=10, fill=ctk.X, expand=True)
group_r_1a.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
# group_r_1b.pack(padx=5, pady=5, fill=ctk.BOTH, expand=True)
group_r_1.pack(padx=5, pady=10, fill=ctk.X, expand=True)
group_r_2a.pack(padx=5, pady=10, fill=ctk.BOTH, expand=True)
group_r_2b.pack(padx=5, pady=10, fill=ctk.BOTH, expand=True)
group_r_2.pack(padx=5, pady=10, fill=ctk.X, expand=True)
left_half.pack(side='left',padx=5, pady=5, fill=ctk.BOTH, expand=True)
right_half.pack(side='left',padx=5, pady=5, fill=ctk.BOTH, expand=True)
combined_halves.pack(padx=10, pady=10, fill=ctk.BOTH, expand=True)
root.after(1, update_plot)
root.mainloop()
# python -m PyInstaller.__main__ --onefile read_sensors.py
main*
0 0
Select folder.
0
Live Share