GNG2101

Design Project User and Product Manual

Athlete Stats

Submitted by:

PerRackFormance, Group B12

Laura Keryakes, 300265134

Yassine Ouloum, 300263213

Jiayi Ma, 300263220

Vivethen Balanchandiran, 300245080

Chelse Rose Vadakkeveettilan Hilariyos, 300214163

December 11, 2022

University of Ottawa

Table of Contents

List of Tables

List of Acronyms and Glossary

Table 1. Acronyms

Acronym	Definition
PRF	PerRackFormance
UPM	User and Product Manual

1 Introduction

During the semester of Fall 2022, the students in the University of Ottawa's Faculty of Engineering have been working closely with various clients. Group B12, also known as PerRackFormance (PRF), had the chance to work on a product that can monitor athletic performance. Among the sports that can be monitored, Group B12 decided to focus on racquet sports, and to test their product specifically on the tennis sport.

The team had their first meeting with the client in which we discussed with the client what were their expectations and specifications concerning the product. They enlisted their expressed needs as well as what the product was expected to do for the client. The client meeting provided us with key information on the product we were tasked to create. Despite our client having no preference on the sport or specific measurement of performance for our project, they provided us with valuable information on how they would like the data obtained to be organized. Our client wanted to be able to properly and efficiently test players over a long period of time to see improvement in their fundamental skills. For example, if we were measuring the speed of a baseball bat swing, our client would want to be able to see the player's best and worst swings over a session along with a graph to display the data in a user-friendly manner. The client also wanted this data to be stored in some sort of database/website so players can easily compare their new tests with old ones to detect improvement in their skills. This way, not only can the data of each player be accessible, the players can also compare their own performance statistics with their teammates.

This User and Product Manual (UPM) provides the information necessary for athletes and coaches of racket sports to effectively use the PerRackFormance (PRF) database and for prototype documentation. The following UPM will provide an overview of the module and will explain exactly how to replicate the prototypes that we have created. We will also ensure how to troubleshoot and activate the panel if there were to be any errors during its creation.

2 Overview

Athletes and coaches playing tennis are looking for a portable, durable and waterproof product that effectively measures their performance and outputs it via a user-friendly platform that allows constant comparison of statistics amongst themselves and their team. This problem is important because nowadays, athletes are only able to receive feedback from their coaches and their teammates, however it is hard to receive quantifiable data of their own performance. In fact, to truly improve in any sport, it is important to be able to see their own metrics related to their performance. After our initial meeting, our client expressed to us their fundamental client needs. From this list, we have interpreted their needs:

Table 1. Interpreted needs of the client alongside relative importance

#	Priority	User needs	Justification for Ranking
1	5	The product detects and collects measurable data related to sports performance.	This is the ultimate goal of the creation of the product which will ultimately lead to the satisfaction of the client.
2	1	The product's materials are sustainable.	The client did not show any preference with the materials of the product.
3	3	The product categorizes data depending on the type of sport techniques.	Though it would be an incredible asset for the product, it would only be added if and only if we can properly respond to higher client needs (e.g., Output the data for the users in an understandable way).

4	5	The product has no effect on the user’s ability to perform.	By hindering the ability for the athlete to perform, results received from the product would instantly be falsified as the athlete would not be performing to the best of their ability.
5	4	The product measures several athletes in a time-efficient manner.	Since we do not know how many athletes will be using our product at the same time, it is important to take that variable into account and ensure that the product can store the data of each athlete without any data-overload or buffering.
6	4	The product outputs sufficient data in an accessible way.	The accessibility of the output of the data is important as it will ensure that our users/client can ultimately understand what they are seeing (i.e., no confusion or queries; data output is straight to the point)
7	4	The product is cost efficient.	Since we have a budget, it is quite important that we realize this cost constraint and that we do not create solutions that are not feasible around that budget.
8	4	The product saves data long term to compare in the future.	One of our client’s biggest desires is for athletes to be able to compare their own data at least three times over the

			season (start, mid, end) to essentially be able to see their progress over time.
9	2	The product is accessible to disabled athletes.	Our client was indifferent if we were to make the product accessible or not to disabled athletes.
10	5	The product covers an athlete's performance range.	The target users are professional players so the product needs to measure performances on that level.
11	3	The product data is centralized.	Although it can become an asset, as it will allow athletes from the same team to compare each other's statistics, the main focus is to be able to output the data of each athlete separately.
12	5	The product is lightweight, portable, durable and waterproof.	This ensures that the product does not hinder the athlete's ability to perform as it won't be in the way. Furthermore, regardless of the strength/performance of the athlete, the product will not break and subsequently falsify data.

Based on the relative importance of each interpreted needs listed above, PRF have decided to mainly focus on providing our client with a product that measures and accurately quantifies data related to sports performance and transfers that data to a website that presents the measurement in a user-friendly and accessible way for users to understand.

What differentiates us from the rest of the tracking products is that our product is attached to the racket rather than on the athlete. This allows a greater range of accessibility for the users. In fact, some users dislike having a tracking device on themselves when they are playing and therefore, having the device on the equipment, allows for these type of athletes to be more inclined to try our product.

Figure 1. Complete setup of PerRackFormance

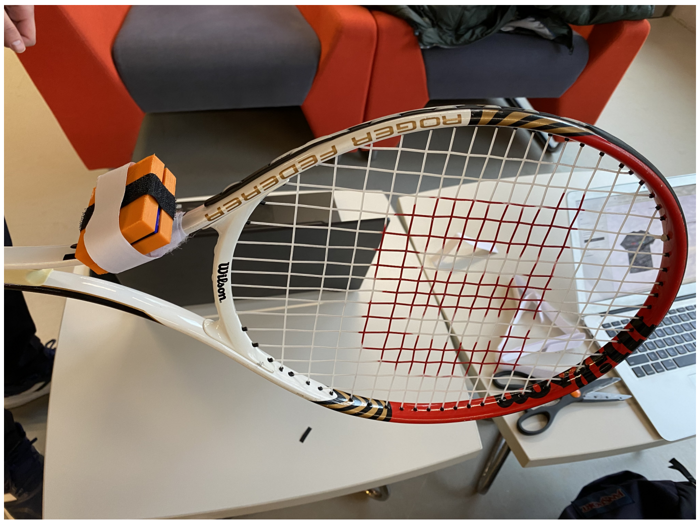


Figure 2. Product when the user receives it.

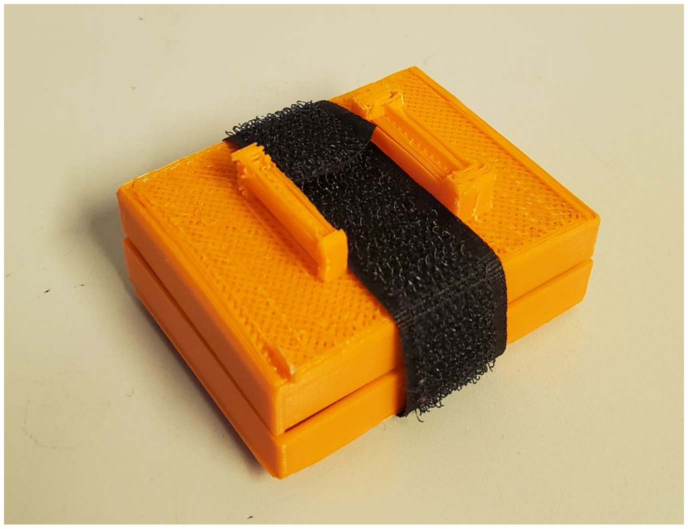
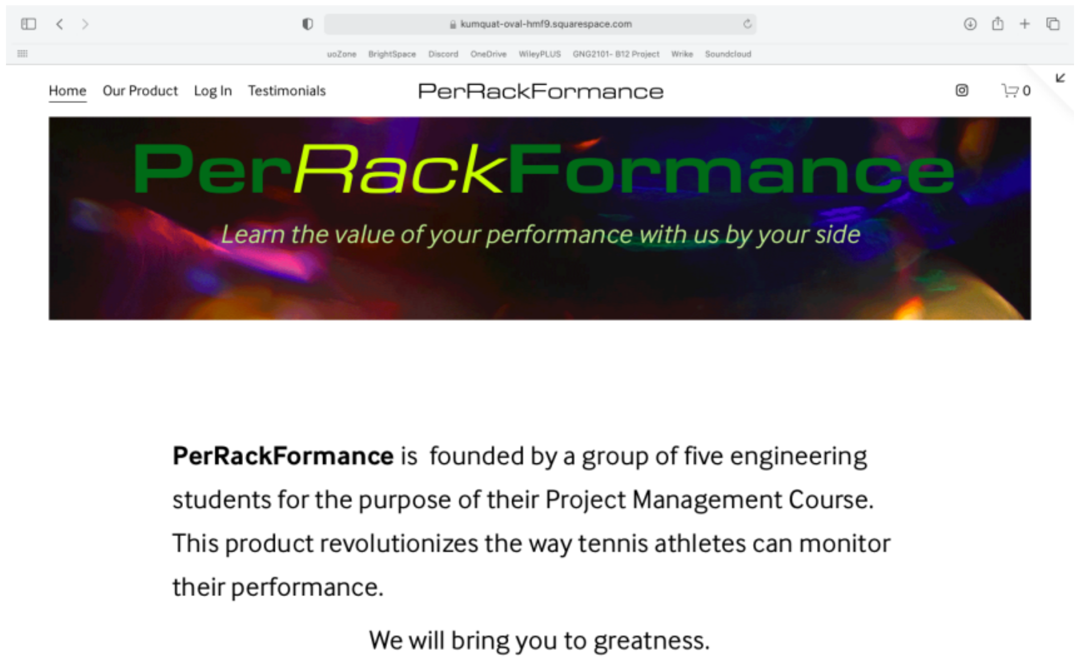
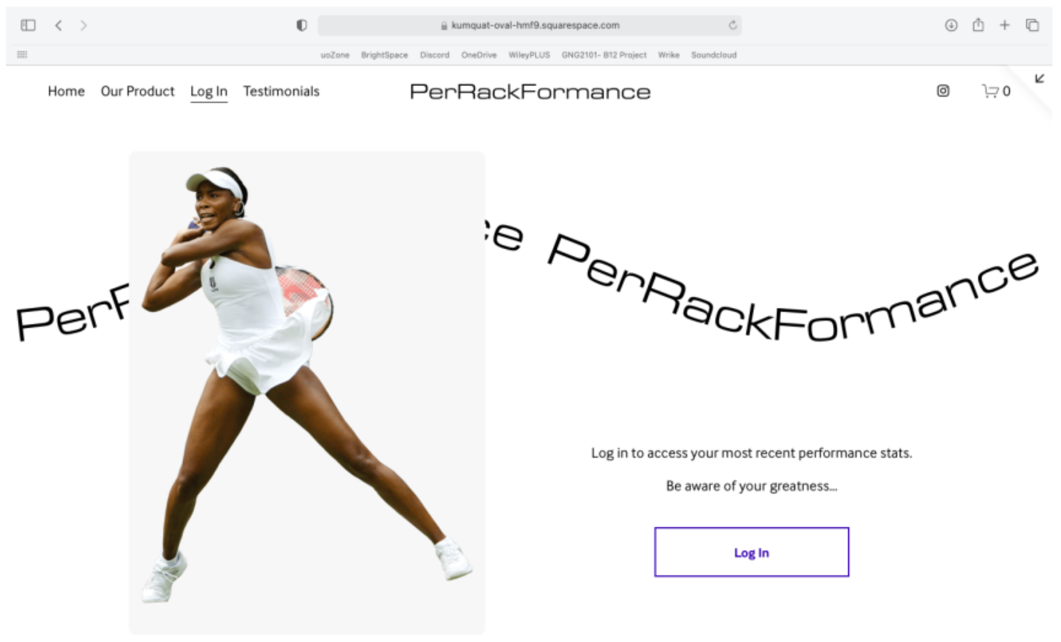


Figure 3-6. Website of PerRackFormance


kumquat-oval-hmf9.squarespace.com
Home
Our Product
Log In
Testimonials
PerRackFormance
0
PerRackFormance
Learn the value of your performance with us by your side

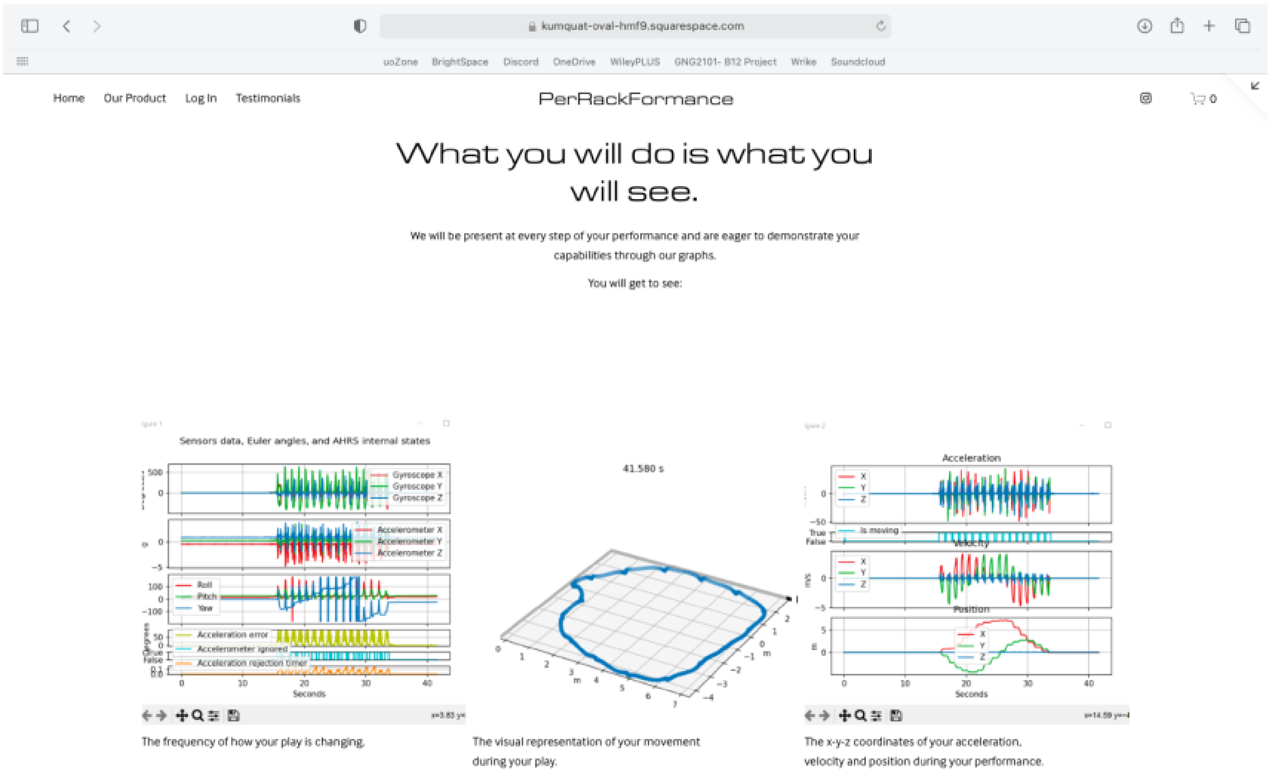
PerRackFormance is founded by a group of five engineering students for the purpose of their Project Management Course. This product revolutionizes the way tennis athletes can monitor their performance.

We will bring you to greatness.



Log in to access your most recent performance stats.

Be aware of your greatness...


kumquat-oval-hmf9.squarespace.com
uoZone BrightSpace Discord OneDrive WileyPLUS GNG2101- B12 Project Wrike Soundcloud
Home Our Product Log In Testimonials
PerRackFormance
Sensors data, Euler angles, and AHRS internal states
Gyroscope X
Gyroscope Y
Gyroscope Z
Accelerometer X
Accelerometer Y
Accelerometer Z
Roll
Pitch
Yaw
Acceleration error
Accelerometer ignored
Acceleration rejection timer
Seconds
41.580 s
Acceleration
Is moving
Velocity
Position
Seconds

What you will do is what you will see.

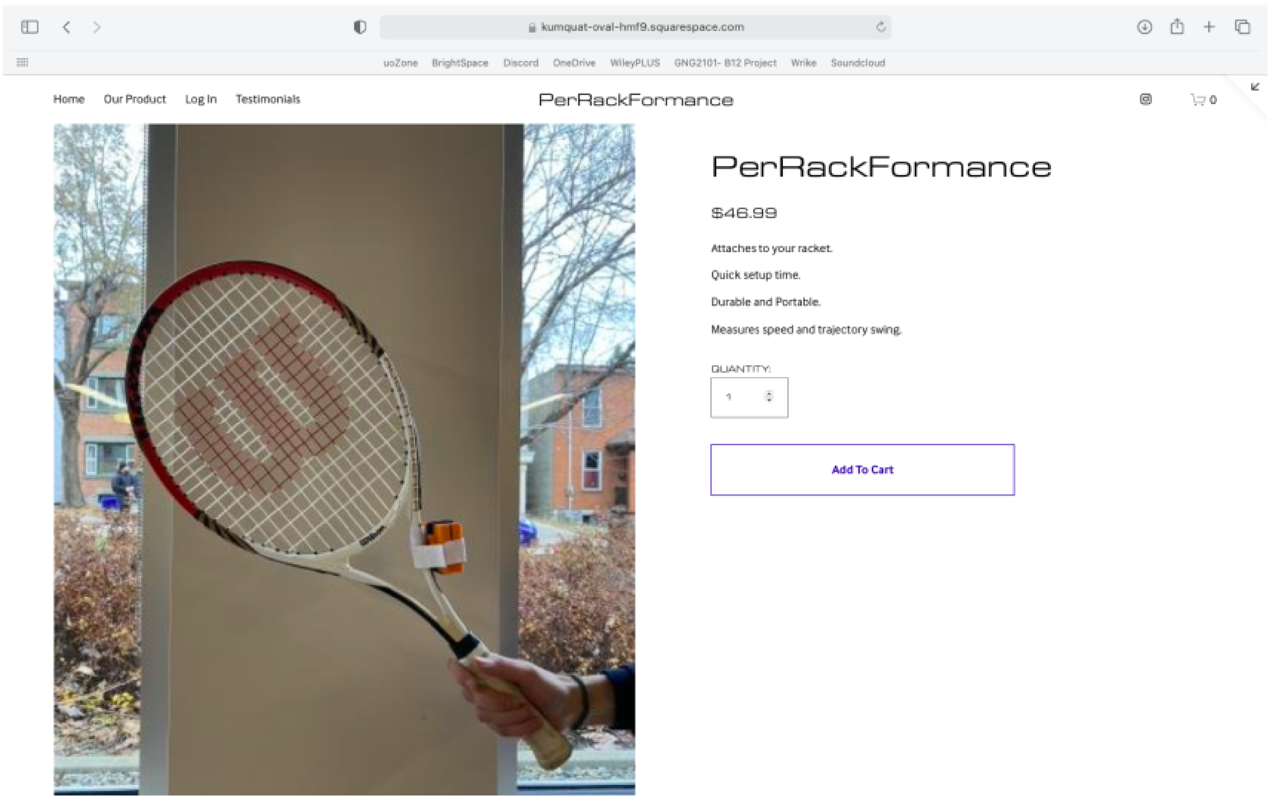
We will be present at every step of your performance and are eager to demonstrate your capabilities through our graphs.

You will get to see:

The frequency of how your play is changing.

The visual representation of your movement during your play.

The x-y-z coordinates of your acceleration, velocity and position during your performance.


kumquat-oval-hmf9.squarespace.com
uoZone BrightSpace Discord OneDrive WileyPLUS GNG2101- B12 Project Wrike Soundcloud
Home Our Product Log In Testimonials
PerRackFormance
QUANTITY:
Add To Cart

PerRackFormance

$46.99

Attaches to your racket.

Quick setup time.

Durable and Portable.

Measures speed and trajectory swing.

The key features of our product includes:

1. The product’s ability to detect and collect data during a performance
2. The product’s ability to categorize and analyze the data based on performance
3. The implementation of a centralized app/website where the user will perceive their stats.

We outlined the tasks to complete each subsystem in a detailed functional decomposition.

Figure 7. High-level decomposition of athlete statistics measurement product.

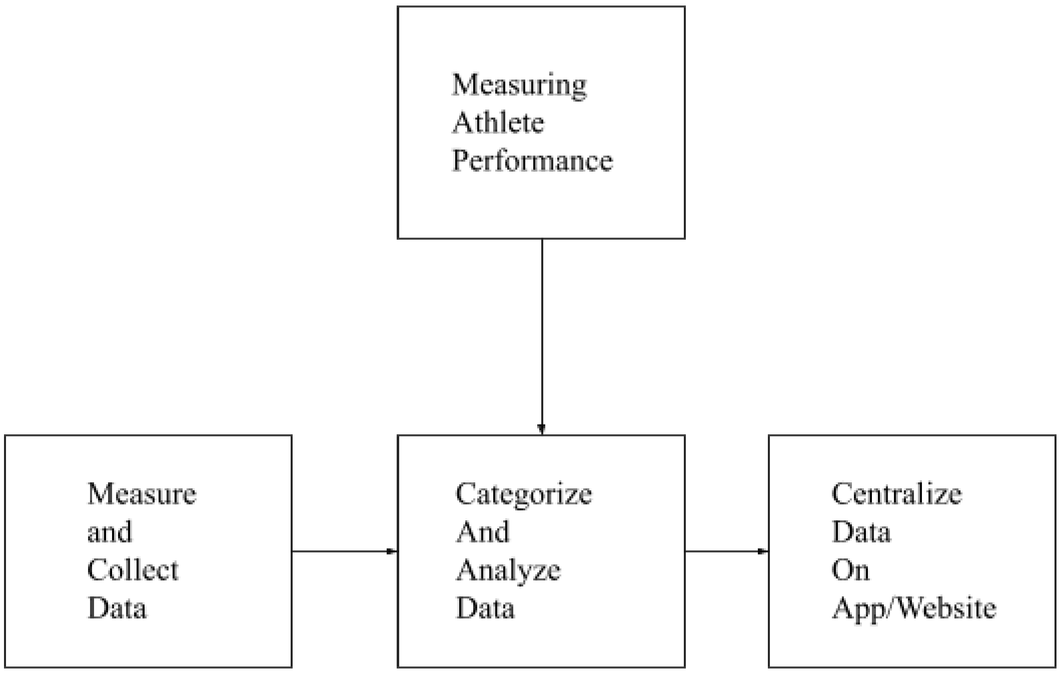

Measuring Athlete Performance
Measure and Collect Data
Categorize And Analyze Data
Centralize Data On App/Website

Figure 8. Detailed functional decomposition of first subsystem

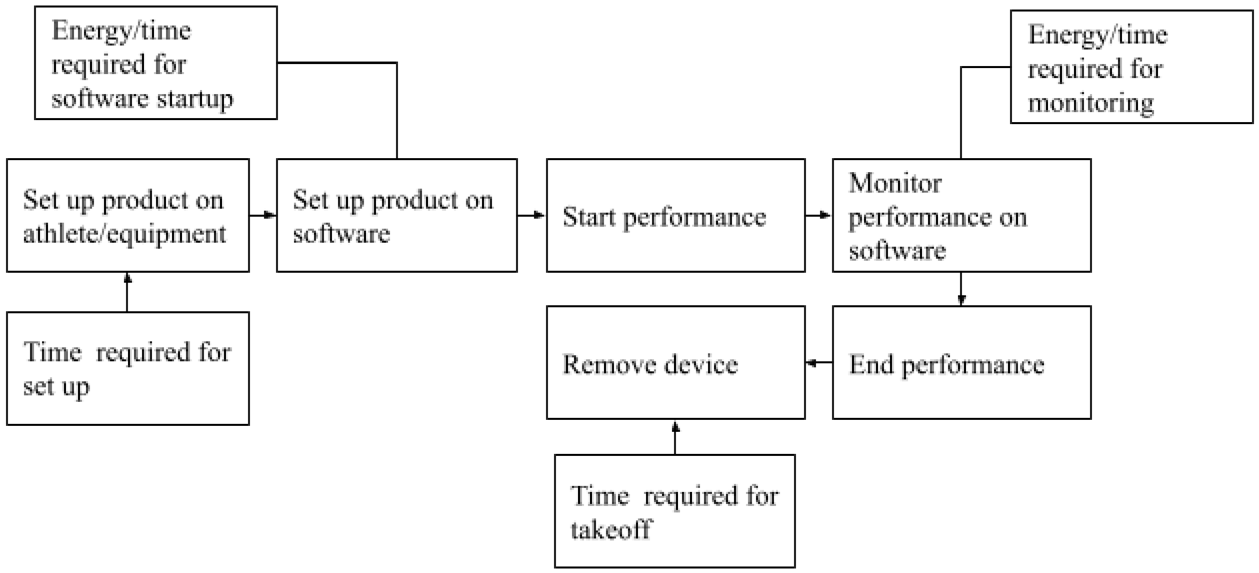
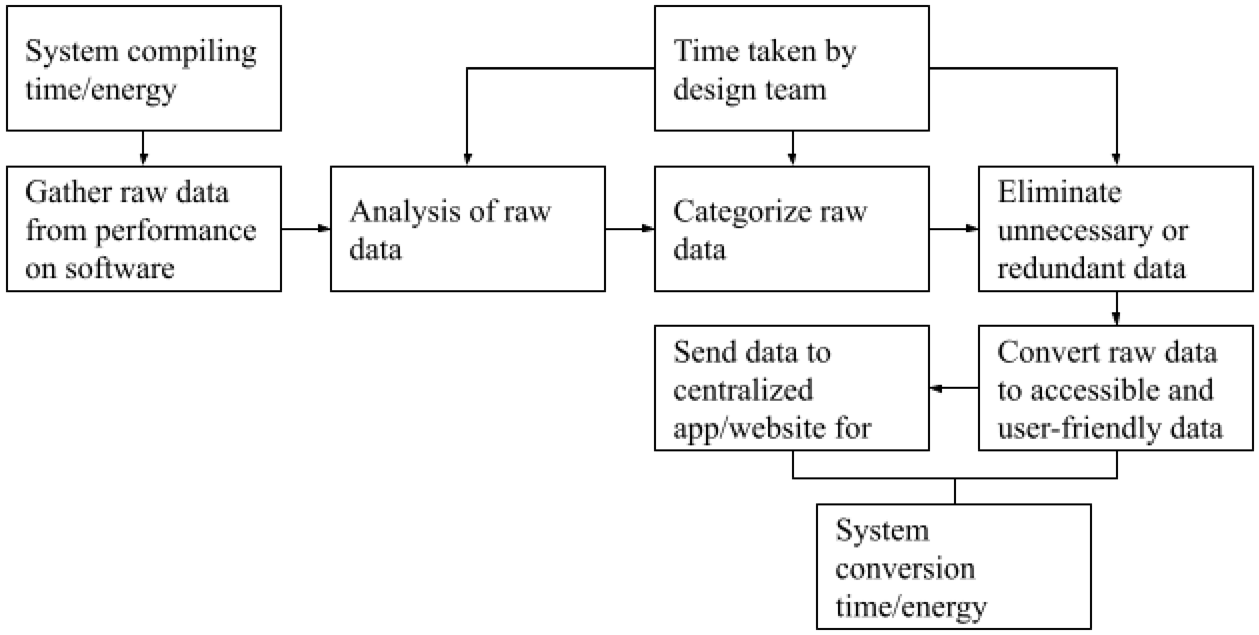

Energy/time required for software startup
Energy/time required for monitoring
Set up product on athlete/equipment
Set up product on software
Start performance
Monitor performance on software
Time required for set up
Remove device
End performance
Time required for takeoff

Figure 9. Detailed functional decomposition of second and third subsystems.


System compiling time/energy
Time taken by design team
Gather raw data from performance on software
Analysis of raw data
Categorize raw data
Eliminate unnecessary or redundant data
Send data to centralized app/website for
Convert raw data to accessible and user-friendly data
System conversion time/energy

Our product consists of a 9V battery, an Arduino Bluetooth chip module and a 6-axis gyroscope and accelerometer; all of which is encased in a 3D-printed case.

2.1 Conventions

No stylistic and command syntax conventions are used within the manual.

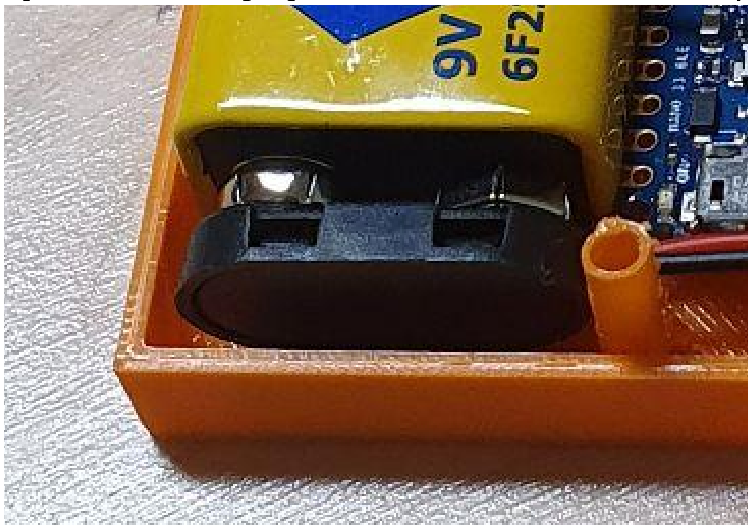
2.2 Cautions & Warnings

It is to note that the current version of our product is not yet as durable as we would like it to be. Therefore, some error margin may come into play when using our product as the case is not yet stable on the racket.

3 Getting started

The product comes in two parts: the **case** and the **board**. To set it up you have to:

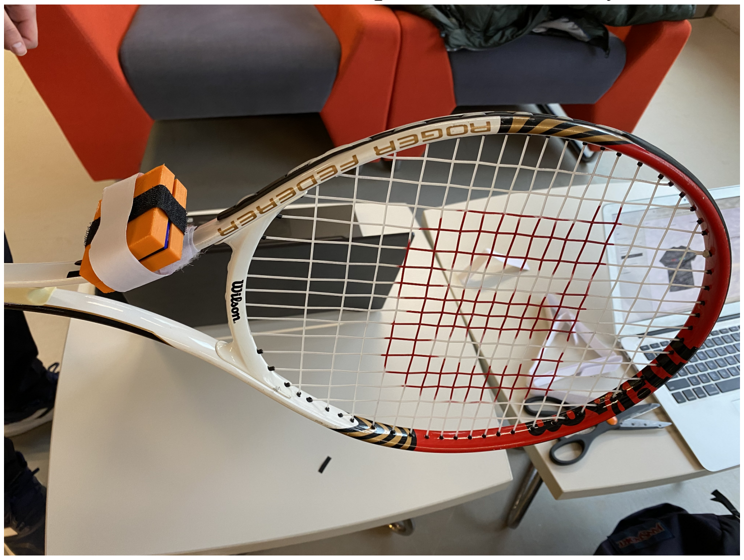
1. Open the **case** and plug the connector head into the battery.



2. Plug the Arduino UNO on the **board** into your computer.
3. Run MS office Excel on your computer.

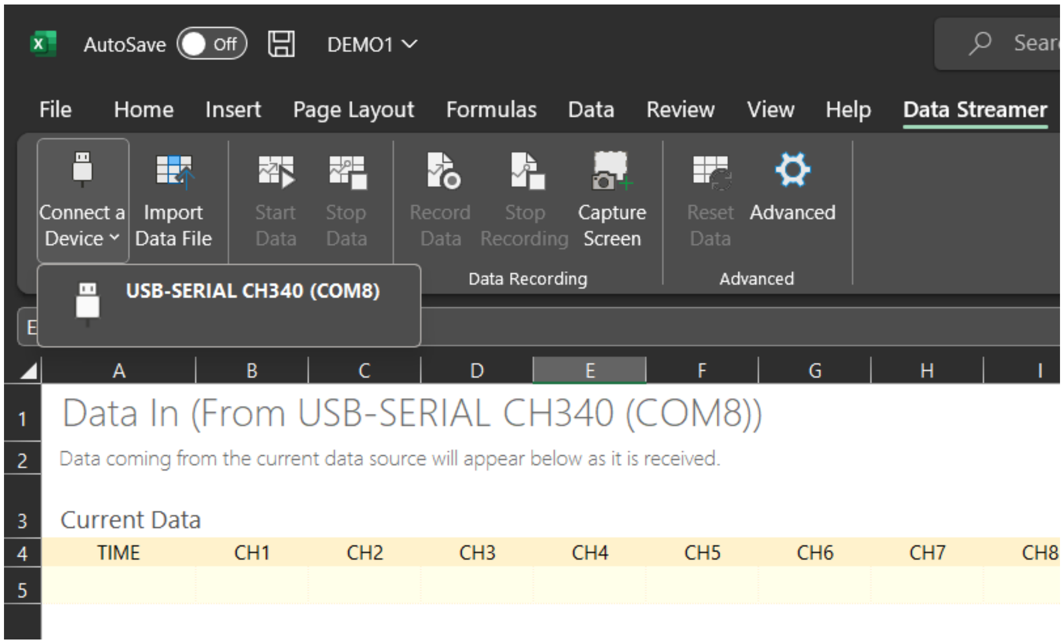
To start collecting data, you have to:

Attach the **case** with the free piece of velcro to your tool.



On an Excel sheet, go to the 'Data Streamer' tab, and click on 'Connect a Device'.

Select "*Name of the device* (COM#)" where # is the port number where you plugged the board.



Data In (From USB-SERIAL CH340 (COM8))

Data coming from the current data source will appear below as it is received.

Current Data

TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8

3.1 Configuration Considerations

The device needs either of the following operating systems to use the 'Data Streamer' add-in: Windows 10, Windows 11. It also needs Microsoft Excel 2016 or more recent versions.

To run the python program, pip command is needed, as well as the libraries: matplotlib, scipy and imufusion. Instructions to download these features are available in section 3.3.

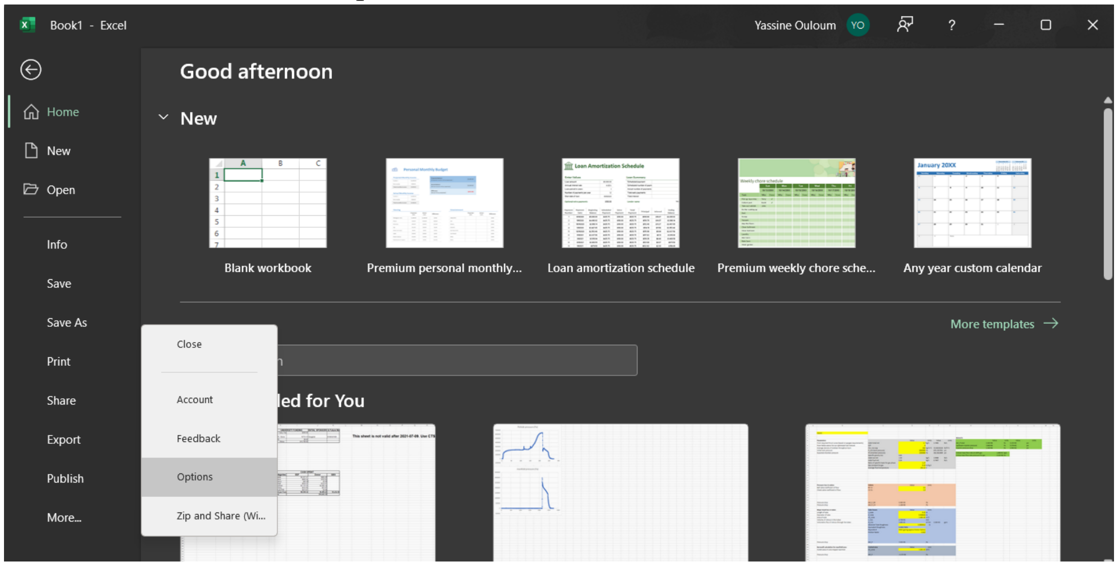
3.2 User Access Considerations

The system can be used by everyone possessing the device and a functional computer with the characteristics depicted in section 3.3.

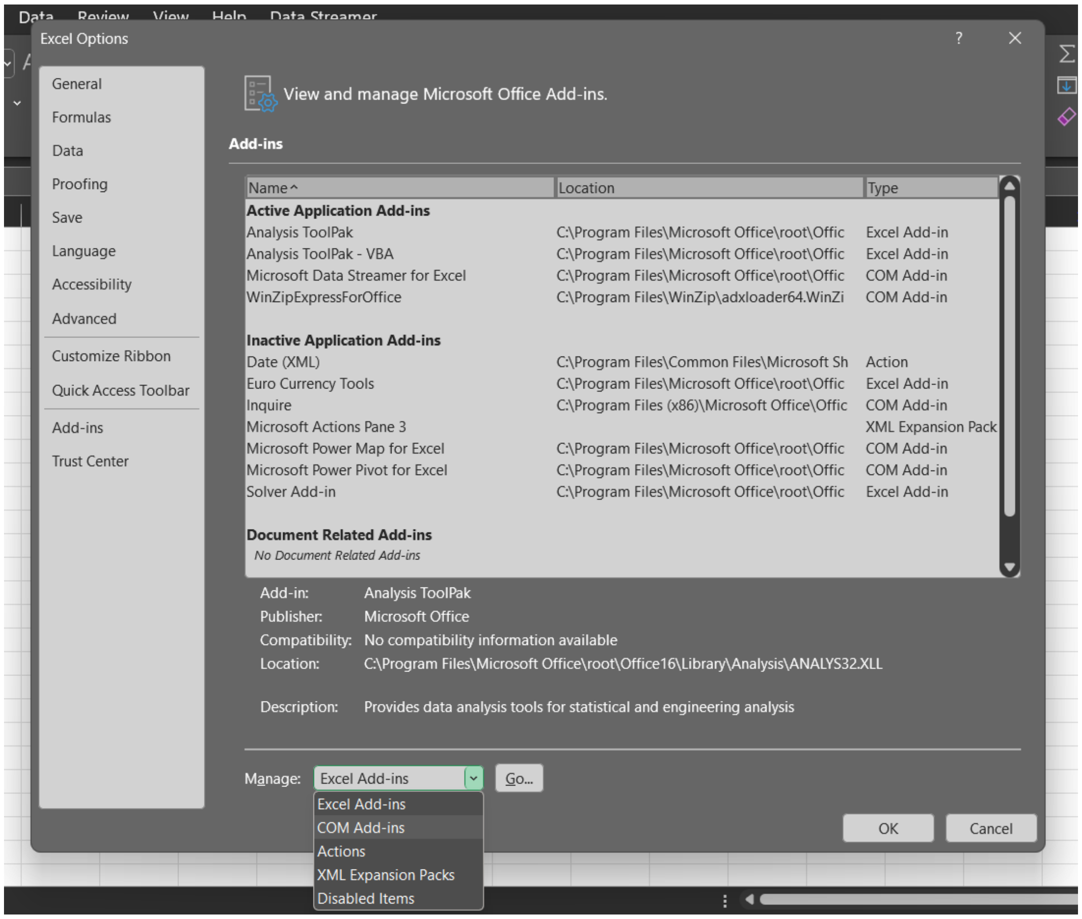
3.3 Accessing/setting-up the System

To download the 'Data Streamer' add-in, the following steps need to be completed:

1. Go to ‘file’ then click on ‘Options’.



2. Select ‘Add-ins’ on the left menu.
3. Click on the drop-down menu next to “Manage” then select ‘COM Add-ins’ and click on ‘Go…’.



View and manage Microsoft Office Add-ins.

Add-ins

Name	Location	Type
Active Application Add-ins		
Analysis ToolPak	C:\Program Files\Microsoft Office\root\Offic	Excel Add-in
Analysis ToolPak - VBA	C:\Program Files\Microsoft Office\root\Offic	Excel Add-in
Microsoft Data Streamer for Excel	C:\Program Files\Microsoft Office\root\Offic	COM Add-in
WinZipExpressForOffice	C:\Program Files\WinZip\adxloader64.WinZi	COM Add-in
Inactive Application Add-ins		
Date (XML)	C:\Program Files\Common Files\Microsoft Sh	Action
Euro Currency Tools	C:\Program Files\Microsoft Office\root\Offic	Excel Add-in
Inquire	C:\Program Files (x86)\Microsoft Office\Offic	COM Add-in
Microsoft Actions Pane 3		XML Expansion Pack
Microsoft Power Map for Excel	C:\Program Files\Microsoft Office\root\Offic	COM Add-in
Microsoft Power Pivot for Excel	C:\Program Files\Microsoft Office\root\Offic	COM Add-in
Solver Add-in	C:\Program Files\Microsoft Office\root\Offic	Excel Add-in
Document Related Add-ins		
No Document Related Add-ins		

Add-in:	Analysis ToolPak
Publisher:	Microsoft Office
Compatibility:	No compatibility information available
Location:	C:\Program Files\Microsoft Office\root\Office16\Library\Analysis\ANALYS32.XLL
Description:	Provides data analysis tools for statistical and engineering analysis

4. Check ‘Microsoft Data Streamer for excel’ and click on ‘OK’.

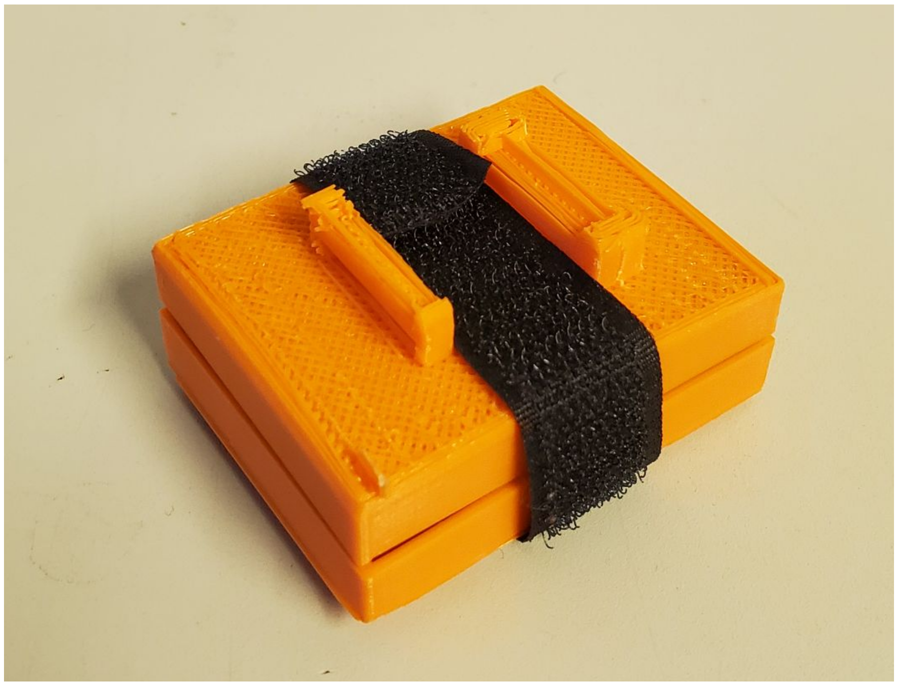
To be able to run the python program provided here, the following steps need to be completed:

1. Download the program.
2. Refer to this guide to install pip: Guide
3. Search for the command prompt.
4. In the command prompt, paste the following lines:
    a. python -m pip install matplotlib --user
    b. python -m pip install scipy --user
    c. python -m pip install imufusion --user

3.4 System Organization & Navigation

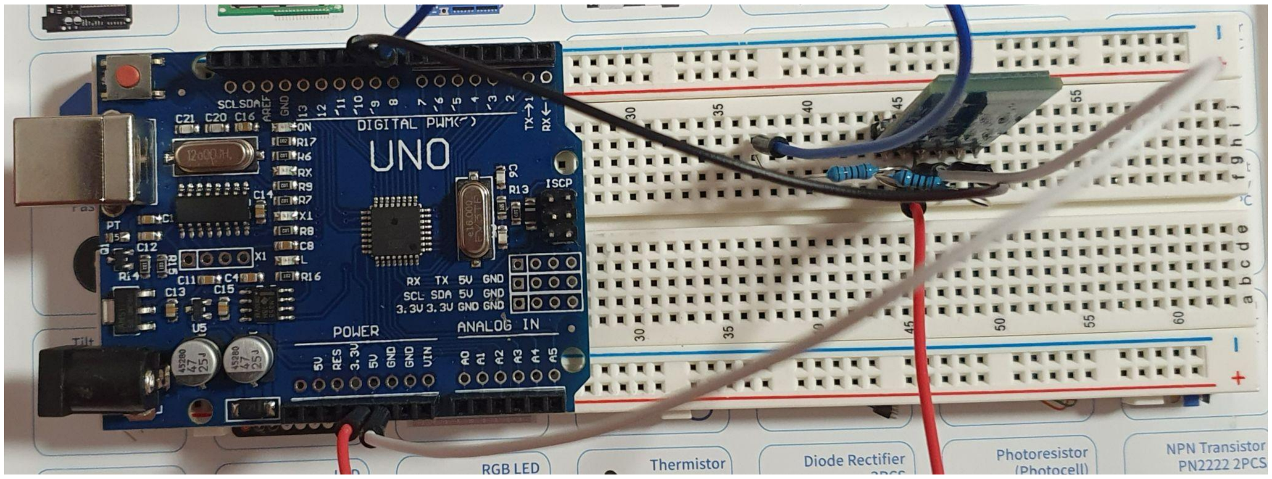
3.4.1 ‘Sending’ component

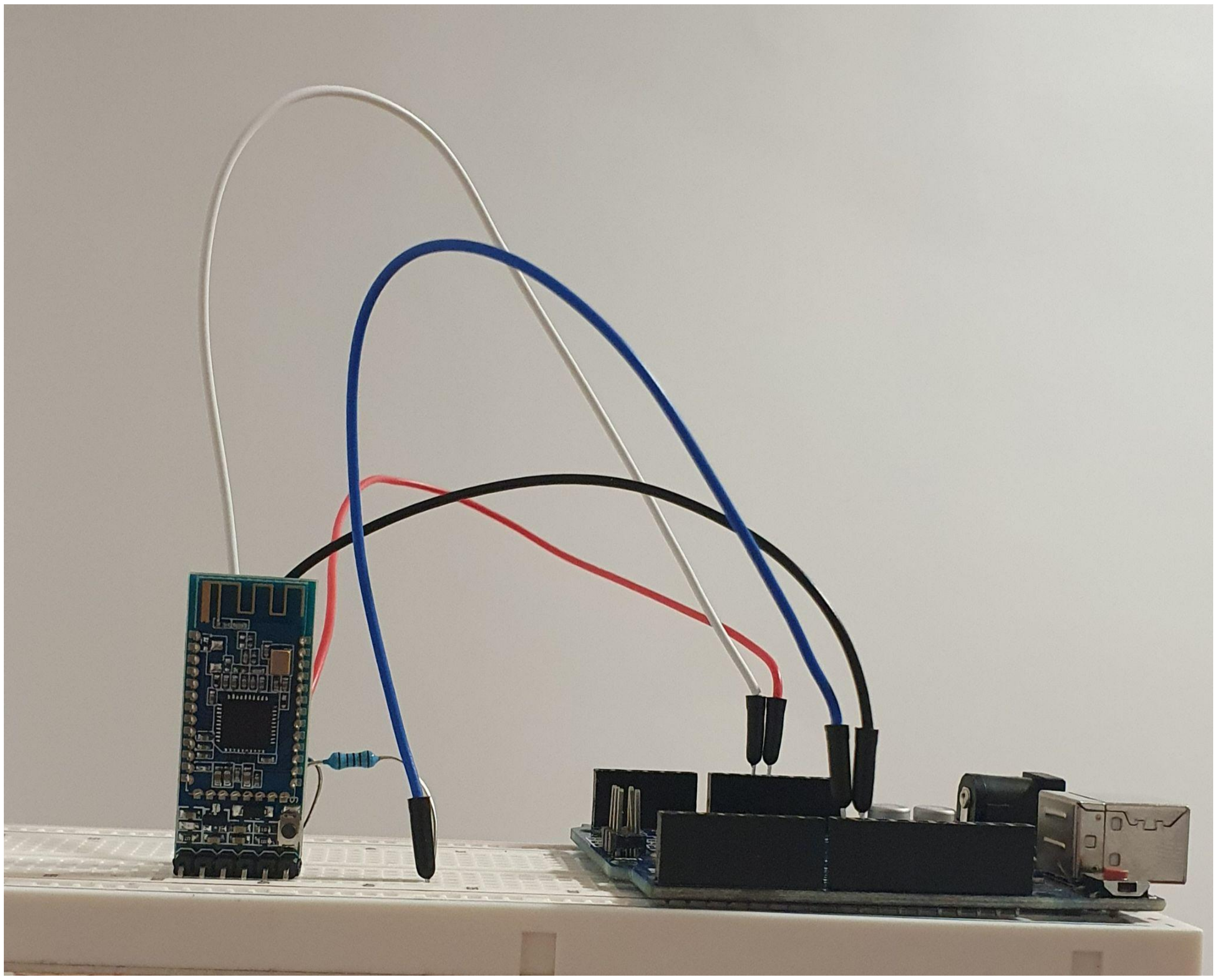
This part is responsible for measuring the data and sending it via BlueTooth. Upon turning on both parts, the connection will be established automatically. It needs input in the form of motion and outputs it (internally) as numerical data.



3.4.2 ‘Receiving’ component

This part is responsible for receiving the data via BlueTooth and displaying it on a serial monitor. Upon turning on both parts, the connection will be established automatically. Receives (internal) input in form of numerical data and outputs it as is.

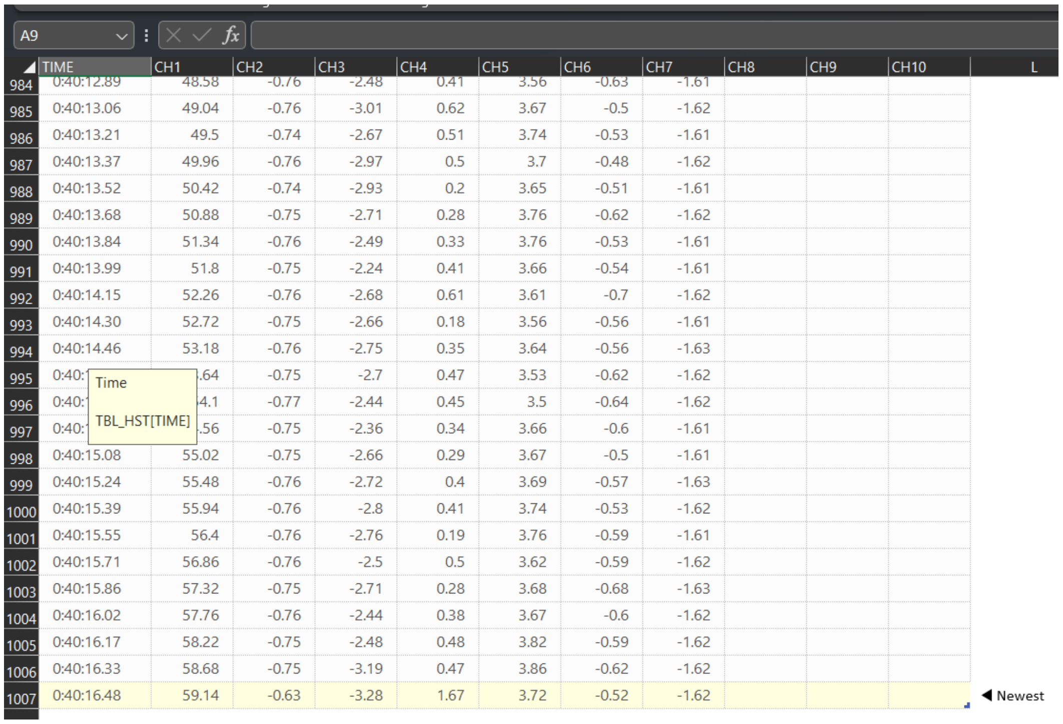




3.4.3 **Software component**

3.4.3.1 Excel:

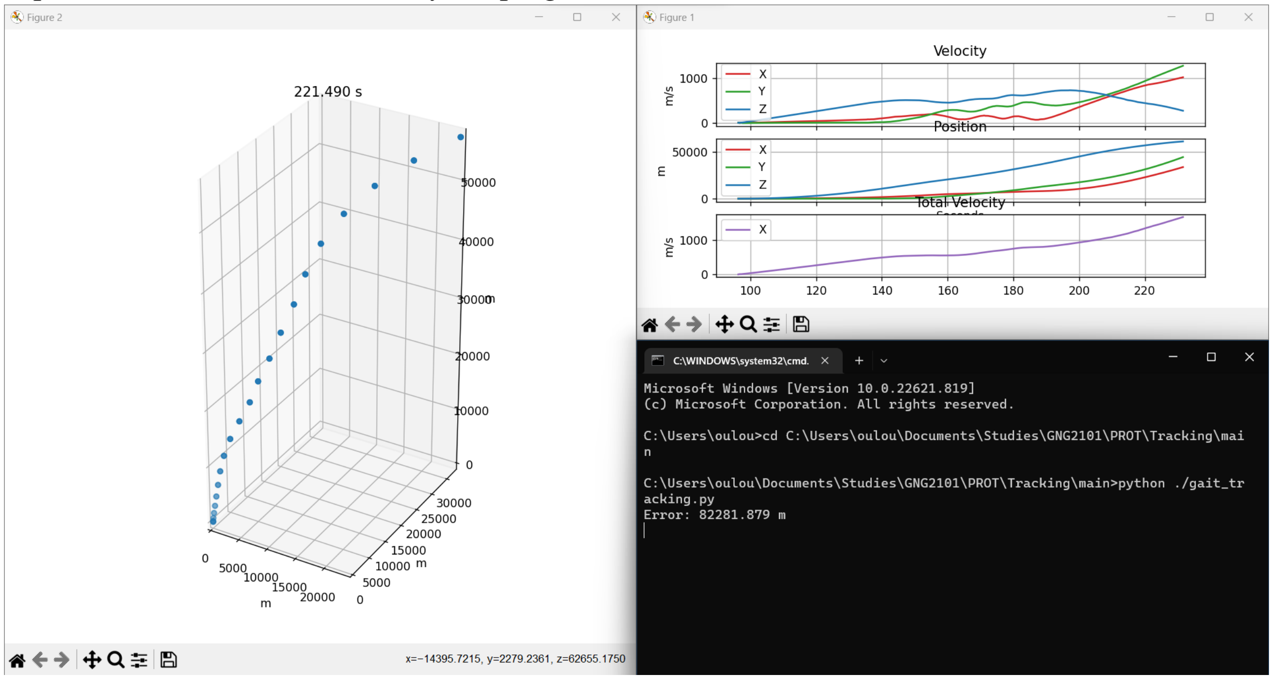
This system transfers the data acquired from the Arduino to the computer. It works via the ‘Data Streamer’ tab in an excel file. It is linked to the ‘Receiving’ component via the computer's COM ports. It receives the numerical data as an input and outputs it as a .csv file.



	TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9	CH10	L
984	0:40:12.89	48.58	-0.76	-2.48	0.41	3.56	-0.63	-1.61				
985	0:40:13.06	49.04	-0.76	-3.01	0.62	3.67	-0.5	-1.62				
986	0:40:13.21	49.5	-0.74	-2.67	0.51	3.74	-0.53	-1.61				
987	0:40:13.37	49.96	-0.76	-2.97	0.5	3.7	-0.48	-1.62				
988	0:40:13.52	50.42	-0.74	-2.93	0.2	3.65	-0.51	-1.61				
989	0:40:13.68	50.88	-0.75	-2.71	0.28	3.76	-0.62	-1.62				
990	0:40:13.84	51.34	-0.76	-2.49	0.33	3.76	-0.53	-1.61				
991	0:40:13.99	51.8	-0.75	-2.24	0.41	3.66	-0.54	-1.61				
992	0:40:14.15	52.26	-0.76	-2.68	0.61	3.61	-0.7	-1.62				
993	0:40:14.30	52.72	-0.75	-2.66	0.18	3.56	-0.56	-1.61				
994	0:40:14.46	53.18	-0.76	-2.75	0.35	3.64	-0.56	-1.63				
995	0:40:	.64	-0.75	-2.7	0.47	3.53	-0.62	-1.62				
996	0:40:1	4.1	-0.77	-2.44	0.45	3.5	-0.64	-1.62				
997	0:40:1	.56	-0.75	-2.36	0.34	3.66	-0.6	-1.61				
998	0:40:15.08	55.02	-0.75	-2.66	0.29	3.67	-0.5	-1.61				
999	0:40:15.24	55.48	-0.76	-2.72	0.4	3.69	-0.57	-1.63				
1000	0:40:15.39	55.94	-0.76	-2.8	0.41	3.74	-0.53	-1.62				
1001	0:40:15.55	56.4	-0.76	-2.76	0.19	3.76	-0.59	-1.61				
1002	0:40:15.71	56.86	-0.76	-2.5	0.5	3.62	-0.59	-1.62				
1003	0:40:15.86	57.32	-0.75	-2.71	0.28	3.68	-0.68	-1.63				
1004	0:40:16.02	57.76	-0.76	-2.44	0.38	3.67	-0.6	-1.62				
1005	0:40:16.17	58.22	-0.75	-2.48	0.48	3.82	-0.59	-1.62				
1006	0:40:16.33	58.68	-0.75	-3.19	0.47	3.86	-0.62	-1.62				
1007	0:40:16.48	59.14	-0.63	-3.28	1.67	3.72	-0.52	-1.62				◀ Newest

3.4.3.2 Python:

This system analyzes the data acquired inputted as a .csv file and outputs information in the form of one 3D graph and 3 2D graphs. It is not linked to the other components, the importation of .csv files is done by the program in each run.



```
Microsoft Windows [Version 10.0.22621.819]
(c) Microsoft Corporation. All rights reserved.

C:\Users\oulou>cd C:\Users\oulou\Documents\Studies\GNG2101\PROT\Tracking\mai
n

C:\Users\oulou\Documents\Studies\GNG2101\PROT\Tracking\main>python ./gait_tr
acking.py
Error: 82281.879 m
```

3.5 Exiting the System

Close the Excel program, command prompt (this action will close all the associated graphs), unplug the Arduino board, open the case and unplug the battery.

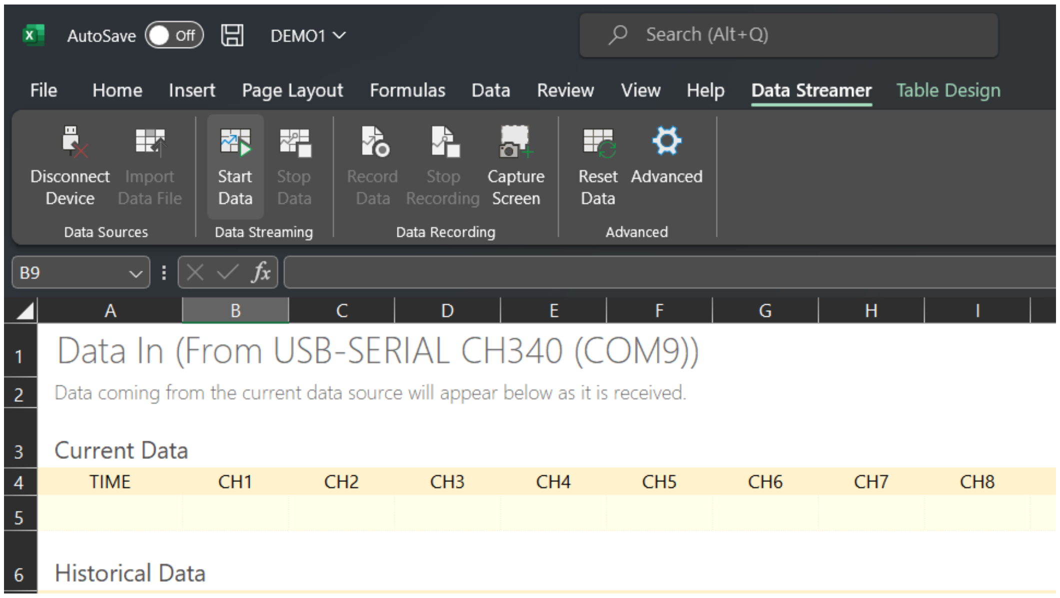
4 Using the System

The following subsections provide detailed, step-by-step instructions on how to record and analyze the data.

4.1 Record Data

In this section, we expect the user to have completed the set-up process described in sections 3.1 to 3.3.

1. On the ‘Data Streamer’ tab in MS Excel, click on Start Data’.



Data In (From USB-SERIAL CH340 (COM9))

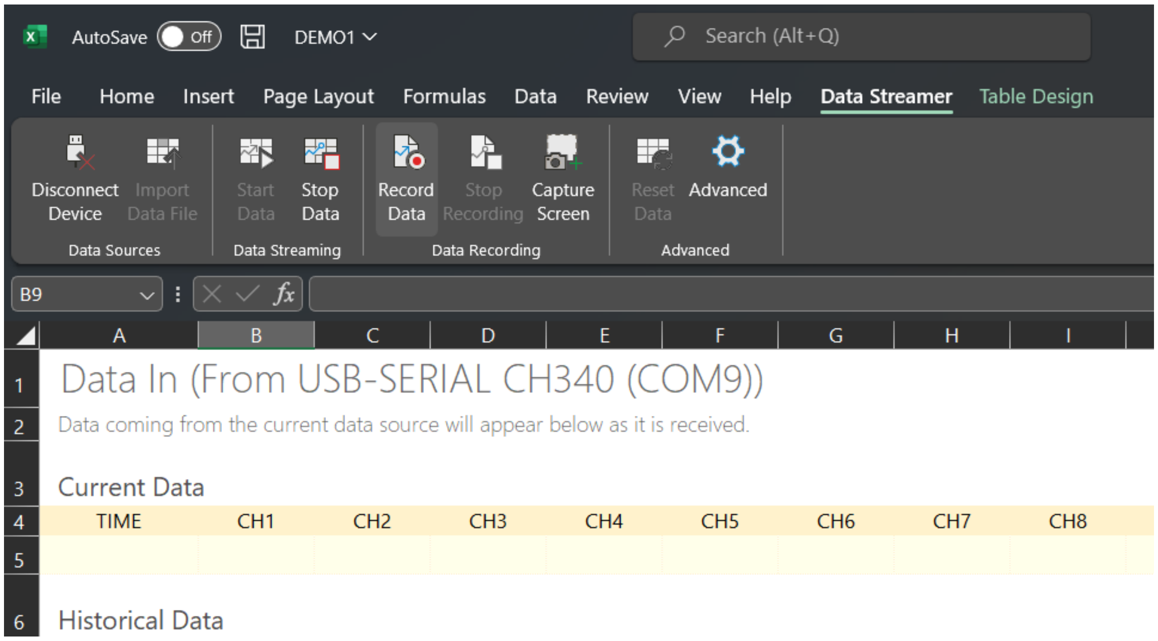
Data coming from the current data source will appear below as it is received.

Current Data

TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8

Historical Data

2. Click on ‘Record Data’.



Data In (From USB-SERIAL CH340 (COM9))

Data coming from the current data source will appear below as it is received.

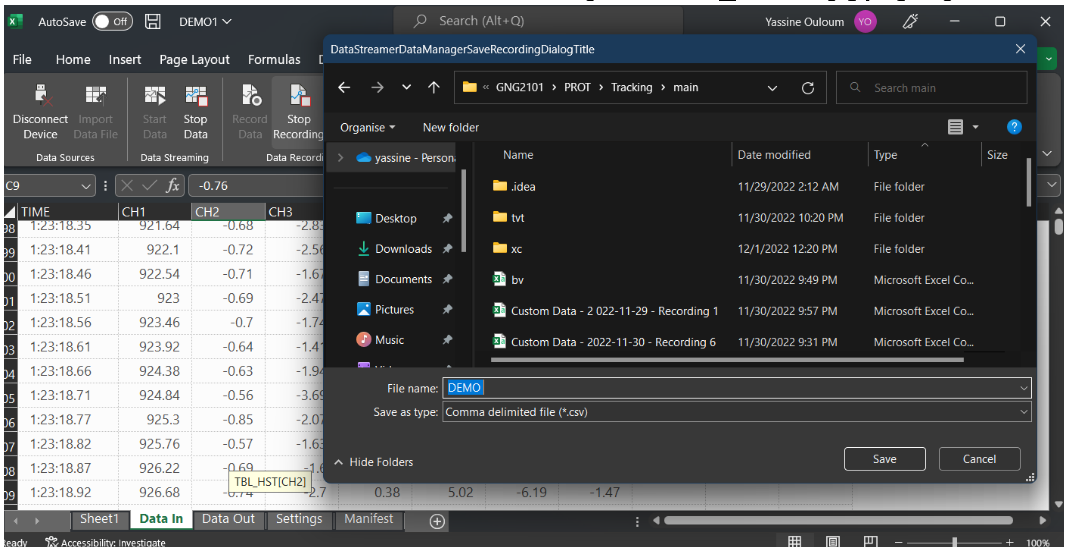
Current Data

TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8

Historical Data

3. Do the motion you wish to analyze.
4. Once you are satisfied with your motion, click on ‘Stop Recording’.

5. Save the .csv file to the same folder containing the ‘Gait_tracking.py’ program.



DataStreamerDataManagerSaveRecordingDialogTitle

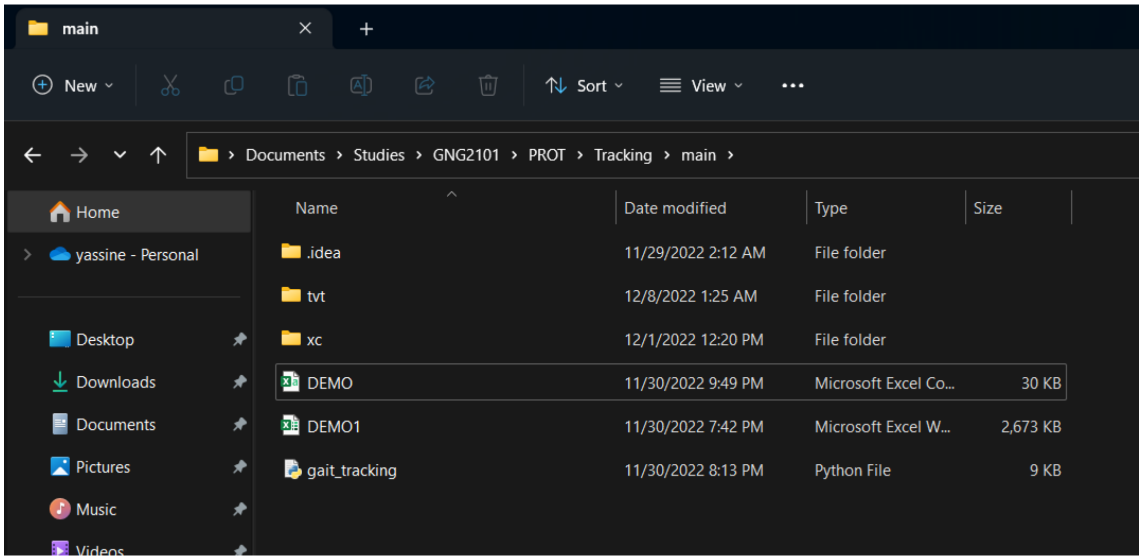
GNG2101 › PROT › Tracking › main

Name	Date modified	Type	Size
.idea	11/29/2022 2:12 AM	File folder	
tvt	11/30/2022 10:20 PM	File folder	
xc	12/1/2022 12:20 PM	File folder	
bv	11/30/2022 9:49 PM	Microsoft Excel Co...	
Custom Data - 2 022-11-29 - Recording 1	11/30/2022 9:57 PM	Microsoft Excel Co...	
Custom Data - 2022-11-30 - Recording 6	11/30/2022 9:31 PM	Microsoft Excel Co...	

File name: DEMO

Save as type: Comma delimited file (*.csv)

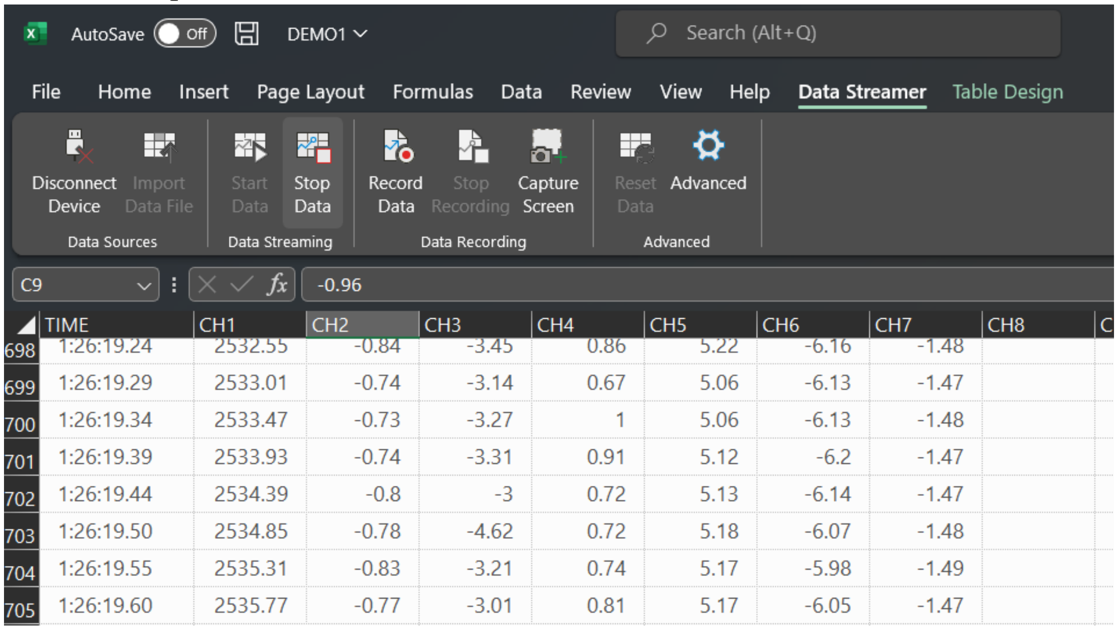
TIME	CH1	CH2	CH3
1:23:18.35	921.64	-0.68	-2.8
1:23:18.41	922.1	-0.72	-2.56
1:23:18.46	922.54	-0.71	-1.67
1:23:18.51	923	-0.69	-2.47
1:23:18.56	923.46	-0.7	-1.74
1:23:18.61	923.92	-0.64	-1.41
1:23:18.66	924.38	-0.63	-1.94
1:23:18.71	924.84	-0.56	-3.69
1:23:18.77	925.3	-0.85	-2.07
1:23:18.82	925.76	-0.57	-1.63
1:23:18.87	926.22	-0.69	-1.6
1:23:18.92	926.68	-0.74	-2.7



Documents › Studies › GNG2101 › PROT › Tracking › main

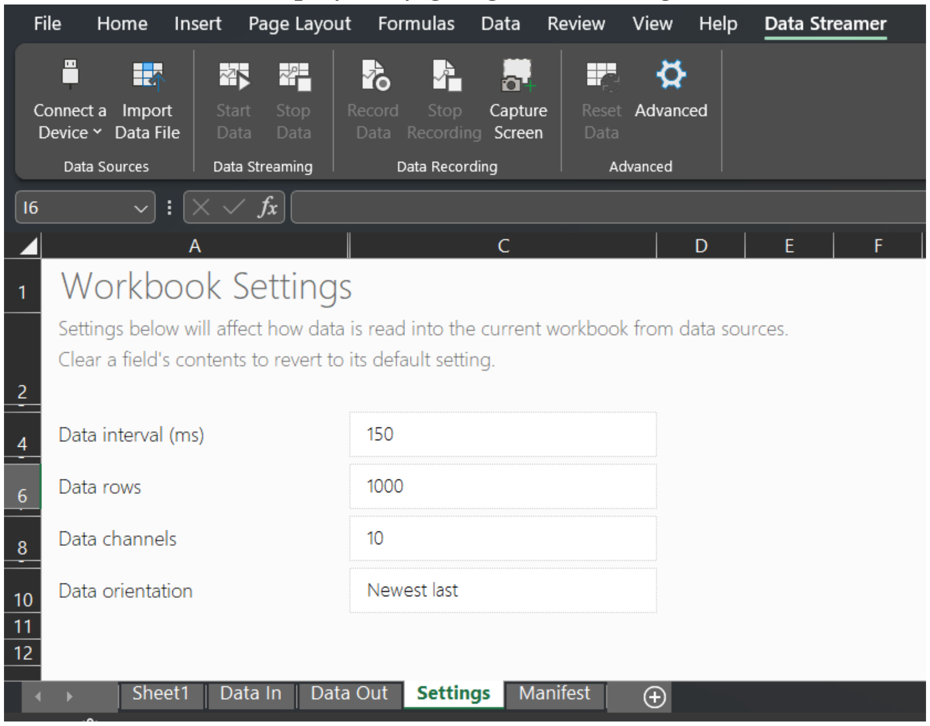
Name	Date modified	Type	Size
.idea	11/29/2022 2:12 AM	File folder	
tvt	12/8/2022 1:25 AM	File folder	
xc	12/1/2022 12:20 PM	File folder	
DEMO	11/30/2022 9:49 PM	Microsoft Excel Co...	30 KB
DEMO1	11/30/2022 7:42 PM	Microsoft Excel W...	2,673 KB
gait_tracking	11/30/2022 8:13 PM	Python File	9 KB

6. Click on 'Stop Data'.



	TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
698	1:26:19.24	2532.55	-0.84	-3.45	0.86	5.22	-6.16	-1.48	
699	1:26:19.29	2533.01	-0.74	-3.14	0.67	5.06	-6.13	-1.47	
700	1:26:19.34	2533.47	-0.73	-3.27	1	5.06	-6.13	-1.48	
701	1:26:19.39	2533.93	-0.74	-3.31	0.91	5.12	-6.2	-1.47	
702	1:26:19.44	2534.39	-0.8	-3	0.72	5.13	-6.14	-1.47	
703	1:26:19.50	2534.85	-0.78	-4.62	0.72	5.18	-6.07	-1.48	
704	1:26:19.55	2535.31	-0.83	-3.21	0.74	5.17	-5.98	-1.49	
705	1:26:19.60	2535.77	-0.77	-3.01	0.81	5.17	-6.05	-1.47	

You can choose to change the rate of data transfer, data orientation, the number of columns, and the number of rows displayed by going to the settings sheet in the workbook.



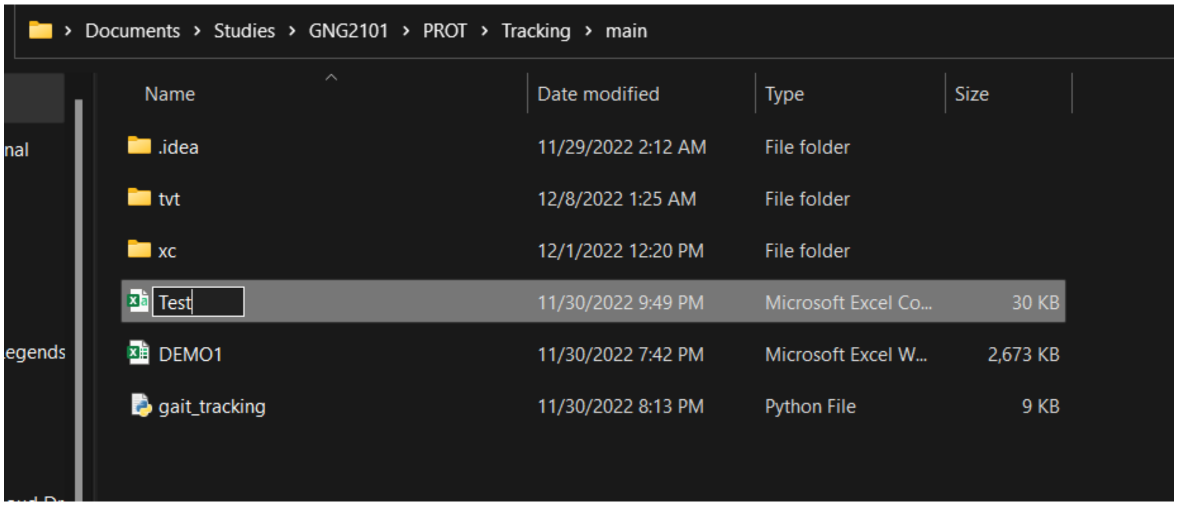
Workbook Settings

Settings below will affect how data is read into the current workbook from data sources.
Clear a field's contents to revert to its default setting.

Data interval (ms)	150
Data rows	1000
Data channels	10
Data orientation	Newest last

4.2 Analyze Data

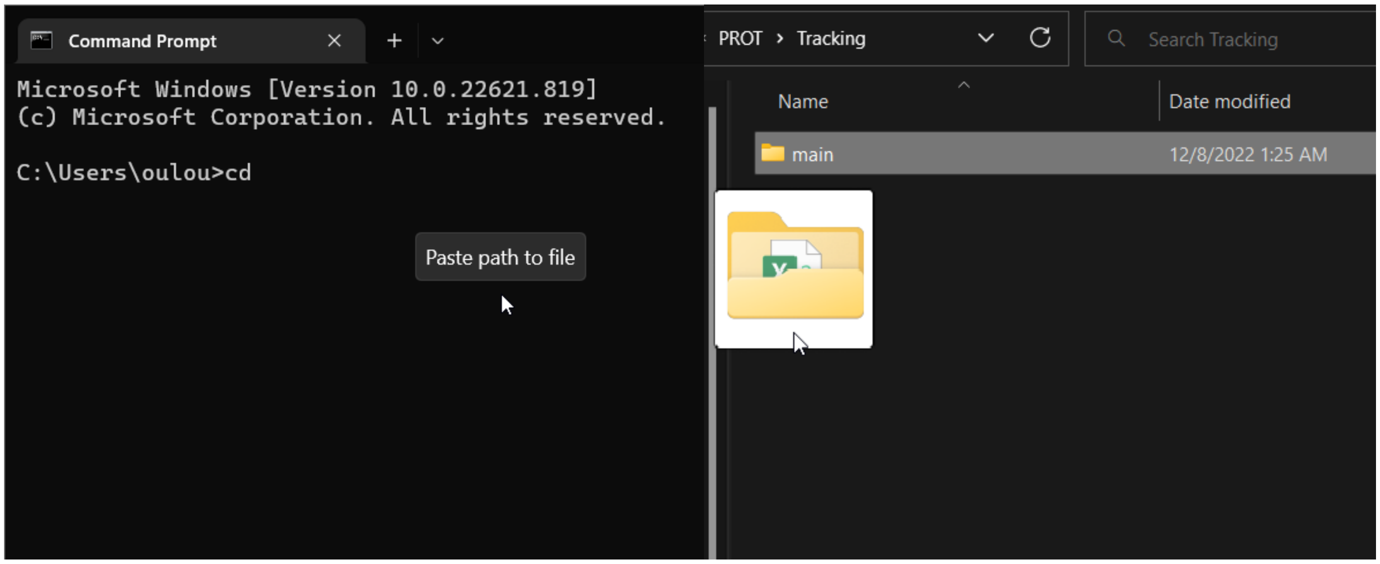
1. Rename the file you want to analyze to “Test”.



Documents › Studies › GNG2101 › PROT › Tracking › main

Name	Date modified	Type	Size
.idea	11/29/2022 2:12 AM	File folder	
tvt	12/8/2022 1:25 AM	File folder	
xc	12/1/2022 12:20 PM	File folder	
Test	11/30/2022 9:49 PM	Microsoft Excel Co...	30 KB
DEMO1	11/30/2022 7:42 PM	Microsoft Excel W...	2,673 KB
gait_tracking	11/30/2022 8:13 PM	Python File	9 KB

2. In the command prompt, type “cd ” then from your file manager, drag the folder containing the python program and your .csv file (Make sure there is a space after

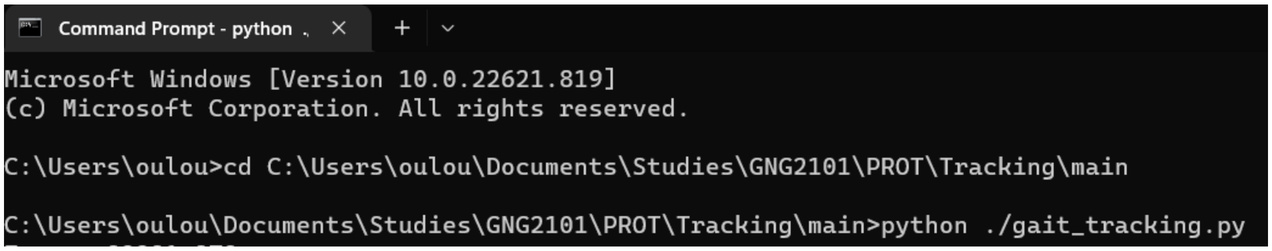


```
Microsoft Windows [Version 10.0.22621.819]
(c) Microsoft Corporation. All rights reserved.

C:\Users\oulou>cd
```

“cd”).

3. Paste the following line: python ./gait_tracking.py



Command Prompt - python .

```
Microsoft Windows [Version 10.0.22621.819]
(c) Microsoft Corporation. All rights reserved.

C:\Users\oulou>cd C:\Users\oulou\Documents\Studies\GNG2101\PROT\Tracking\main

C:\Users\oulou\Documents\Studies\GNG2101\PROT\Tracking\main>python ./gait_tracking.py
```

5 Troubleshooting & Support

The following subsections indicate the actions to take when dealing with a problem related to our product.

5.1 Maintenance

The main form of maintenance required is to replace the battery inside the case if it runs out of power. A good indication to tell when the battery is out of power is by checking if the light inside of the case is turned on when in use. If a light appears then the battery works fine; otherwise, replace it with a new one for further use.

5.2 CSV File Issues

There are many potential problems that could cause the CSV file to not be created. Here are some simple checks to see what is potentially causing the problem.

5.2.1 Arduino Checks

The following are two main ways to check if the arduino is the root cause for the CSV file issue.

5.2.1.1 Restarting the Arduino:

A potential cause of the problem could be the arduino. Here is a simple procedure to attempt a swift fix for the problem. Try unplugging and replugging the arduino into the computer or laptop. Restart the arduino using the arduino restart button.

5.2.1.2 Checking if the Arduino is Functional:

A rare possibility is that the arduino may not be working at all. To check this, try connecting to it using a mobile device by turning on bluetooth and scanning for all devices. If the arduino does not appear in the list then the arduino bluetooth connection is the issue. If this is the case, follow section 5.4 for instructions on how to contact customer support for further assistance on replacing the faulty component.

5.2.2 Computer & Software Checks

The following are two main ways to check if the computer or software is the root cause for the CSV file issue.

5.2.2.1 Restarting the Computer & Software:

A potential cause of the problem could be the computer or the software used to run the arduino. Here is a simple procedure to attempt a swift fix for the problem. Try closing and reopening the arduino software on the computer or laptop. Restart the computer or try connecting to a different one.

5.2.2.2 Checking the Port Number Connected:

An easily missed mistake to make is attempting to connect excel to arduino using the incorrect port number. Make sure you are connected to the correct one, and try again.

5.2.3 Checking the Wiring

A rare, but possible problem could be related to the inner wiring within the 3D printed case. First, check that the lights on the Arduino nano and MPU6050 are on and high after connecting the battery. If this is the case, follow section 5.4 for instructions on how to contact customer support for further assistance in replacing the faulty component.

5.3 3D Graph Issues

Firstly, make sure to follow the steps indicated in section 3.3 of the user manual. If the graph is still not working, the csv file may be corrupted causing the problem. Try another swing with the racquet and see if a graph appears when running the python script. If no graph appears, follow section 5.4 for instructions on how to contact customer support for further assistance.

5.4 Support

If any issues arise that are unable to be solved using the troubleshooting methods above, feel free to contact us using our email: "PerRackformance_HelpCenter@gmail.com", or via phone call using the following number: "613-456-8690". Our in person help center is also available during all working days (Monday - Friday) from 8:30 AM - 10:00 PM at "24 fake street" in Ottawa Ontario. We welcome anyone to ask questions or assistance related to our product.

6 Product Documentation

The final prototype is a device that tracks measurements to analyze tennis player data, which consists of two subsystems: sensing equipment(mechanical) and output software.

Regarding the sensing device (mechanical) part, the device consists of a chip, wires, and a case. The team chose to use the Arduino nano BLE and MPU6050 chips as the core of the data transmission because of the small size, light weight, and low cost of the Bluetooth chip. Due to the maturity and efficiency of 3D printing technology, the team finally chose to use this technology to make the housing of the device. The 3D printer and plastic material was used to create the case for the chip and wires. The device will be mounted on a tennis racket handle and will not interfere with the player's normal swinging motion.

For the output software part, the team chose Arduino and Python as the main writing software. Since Python has the most extensive drawing library for most basic drawing needs, the team used it as a part of the software output. All the data collected from the sensing devices are transferred to a csv file via Bluetooth and analyzed and transformed into visual images by the python program, which is finally output to the web client.

6.1 Case

6.1.1 BOM (Bill of Materials)

- PLA filament (free via uni)
- velcro (1.25 CAD in dollarama)
- Ultimaker Cura (free)

6.1.2 Equipment list

- 3D printer
- scissors (cut velcro etc.)
- SD card (borrowed from uni)
- SD card to usb key (borrowed from uni)

6.1.3 Instructions

- convert 3D part to .stl format in solidworks.
- install cura, import .stl project, slice it, select 0.25 mm nozzle size, add supports for the top part of the case.
- download the stl file in the SD card
- put the SD card in the printer
- redo the steps for the bottom case

6.2 ‘Sending’ component

6.2.1 BOM (Bill of Materials)

- arduino nano BLE
- MPU6050
- battery
- connecter head (to connect the battery)
- wires

6.2.2 Equipment list

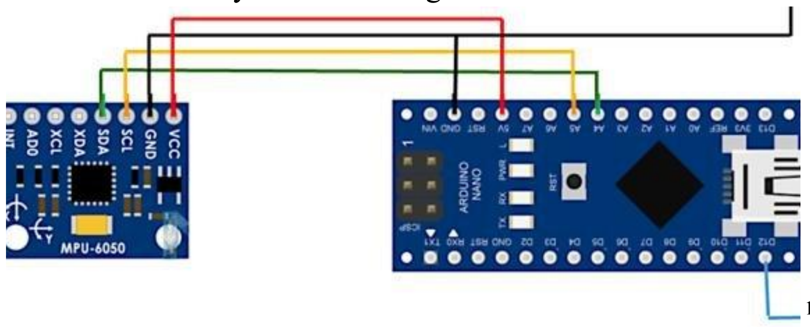
- soldering iron
- solder
- pliers
- wire cutters

6.2.3 Instructions

-upload the code (appendix 2) in the arduino nano

-cut the wires to approximately 4cm each.

-wire the system following this schematic



[1]

-solder everything

-solder the free parts of the connector head to the v in and and pins

6.3 ‘Receiving’ component

6.3.1 BOM (Bill of Materials)

- 1k and 2k Ohm resistors

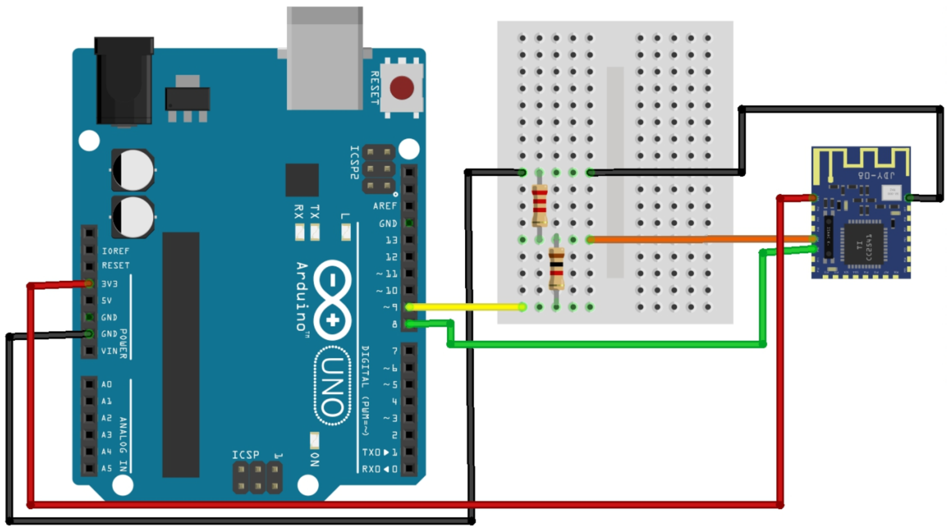
[1] https://www.researchgate.net/figure/Connections-of-Arduino-Nano-board-to-MPU6050-module-and-laptop_fig4_353674719

- HM10 bluetooth module
- breadboard
- arduino UNO
- wires

6.3.2 Equipment list

6.3.3 Instructions

- upload the code (appendix 2) in the arduino uno
- wire everything like this



[2]

6.4 Software component

6.4.1 BOM (Bill of Materials)

- Excel
- python program

6.4.2 Equipment list

- pc

6.4.3 Instructions

- download everything, set up is explained in section 3

[2] https://handiko.github.io/JDY-08-Reflash/

6.5 Testing & Validation

To assure this machine functions effectively and safely, our team continue three different tests on the final prototype include:

1. Arduino program testing

   a.Whether the program works properly

   b.Whether the data from the sensor can be imported into a csv file

2. Python visualization program testing

   a.Whether the library is installed

   b.Whether the original data (acceleration, Rotation) is converted into velocity and location

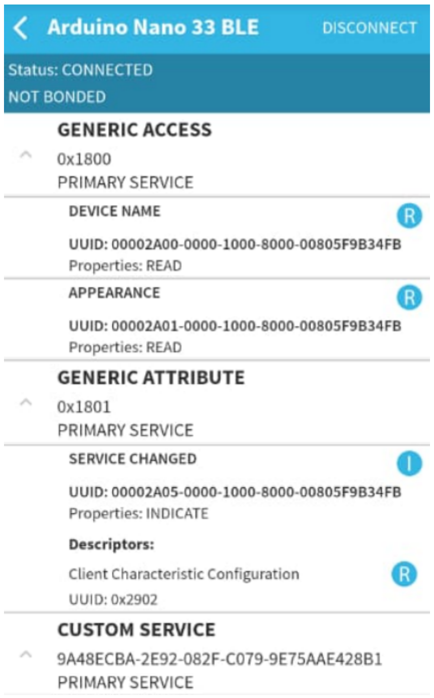
   c.Whether to convert the data into images

3. Case stability testing

   a.Whether the housing can be loaded with all chips

   b.Whether the housing can be mounted on the racket

For test 1, the program runs successfully and outputs the acceleration XYZ and rotation XYZ to the phone application "BLE Scanner" for testing. Finally, the code was added to input the data into a csv file.



Status: CONNECTED
NOT BONDED

GENERIC ACCESS
0x1800
PRIMARY SERVICE

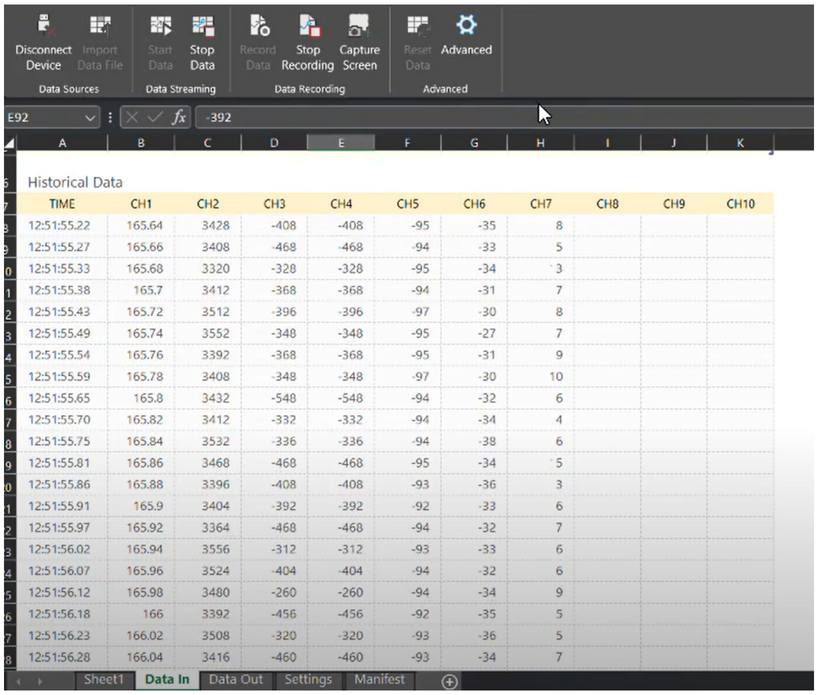
DEVICE NAME
UUID: 00002A00-0000-1000-8000-00805F9B34FB
Properties: READ

APPEARANCE
UUID: 00002A01-0000-1000-8000-00805F9B34FB
Properties: READ

GENERIC ATTRIBUTE
0x1801
PRIMARY SERVICE

SERVICE CHANGED
UUID: 00002A05-0000-1000-8000-00805F9B34FB
Properties: INDICATE
Descriptors:
Client Characteristic Configuration
UUID: 0x2902

CUSTOM SERVICE
9A48ECBA-2E92-082F-C079-9E75AAE428B1
PRIMARY SERVICE



-392

Historical Data

TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9	CH10
12:51:55.22	165.64	3428	-408	-408	-95	-35	8			
12:51:55.27	165.66	3408	-468	-468	-94	-33	5			
12:51:55.33	165.68	3320	-328	-328	-95	-34	3			
12:51:55.38	165.7	3412	-368	-368	-94	-31	7			
12:51:55.43	165.72	3512	-396	-396	-97	-30	8			
12:51:55.49	165.74	3552	-348	-348	-95	-27	7			
12:51:55.54	165.76	3392	-368	-368	-95	-31	9			
12:51:55.59	165.78	3408	-348	-348	-97	-30	10			
12:51:55.65	165.8	3432	-548	-548	-94	-32	6			
12:51:55.70	165.82	3412	-332	-332	-94	-34	4			
12:51:55.75	165.84	3532	-336	-336	-94	-38	6			
12:51:55.81	165.86	3468	-468	-468	-95	-34	5			
12:51:55.86	165.88	3396	-408	-408	-93	-36	3			
12:51:55.91	165.9	3404	-392	-392	-92	-33	6			
12:51:55.97	165.92	3364	-468	-468	-94	-32	7			
12:51:56.02	165.94	3556	-312	-312	-93	-33	6			
12:51:56.07	165.96	3524	-404	-404	-94	-32	6			
12:51:56.12	165.98	3480	-260	-260	-94	-34	9			
12:51:56.18	166	3392	-456	-456	-92	-35	5			
12:51:56.23	166.02	3508	-320	-320	-93	-36	5			
12:51:56.28	166.04	3416	-460	-460	-93	-34	7			

For test 2, the team chose to use pip with matplotlib as a library to visualize the raw data and successfully transform the velocity and location.

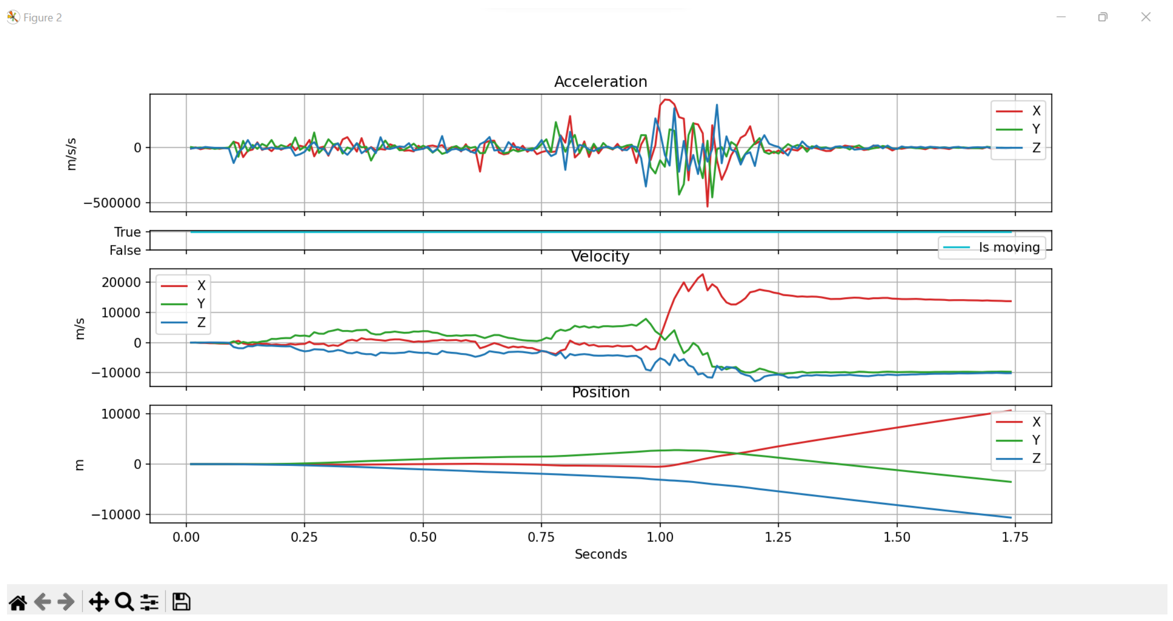


Figure 2
Acceleration
m/s/s
0
−500000
X
Y
Z
True
False
Is moving
Velocity
20000
10000
0
−10000
m/s
X
Y
Z
Position
10000
0
−10000
m
X
Y
Z
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
Seconds

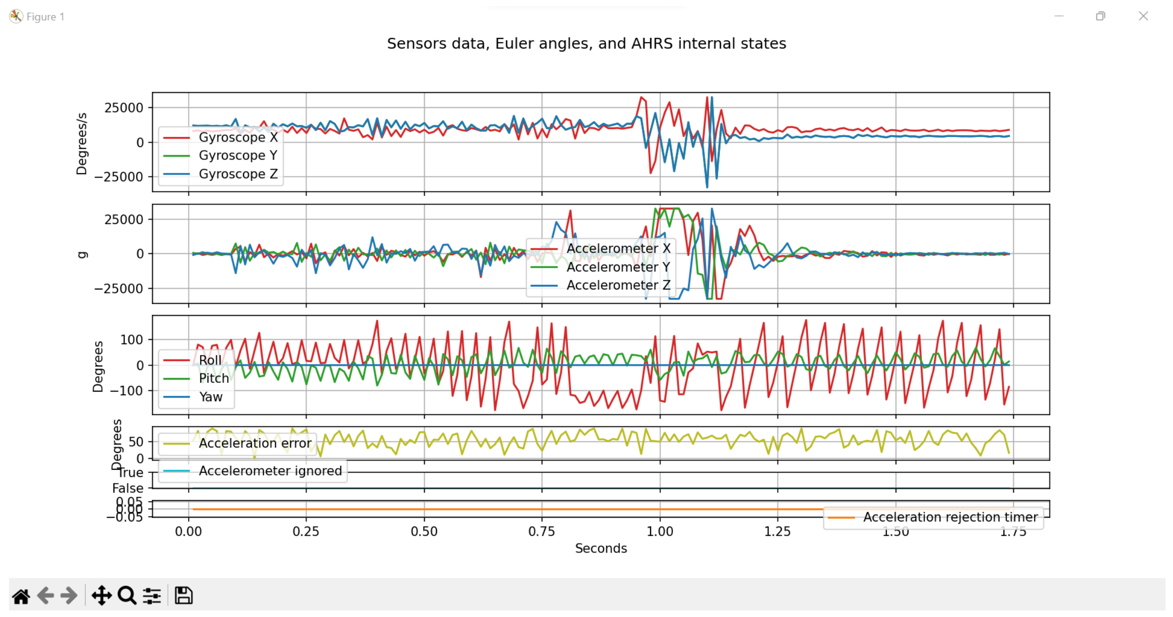
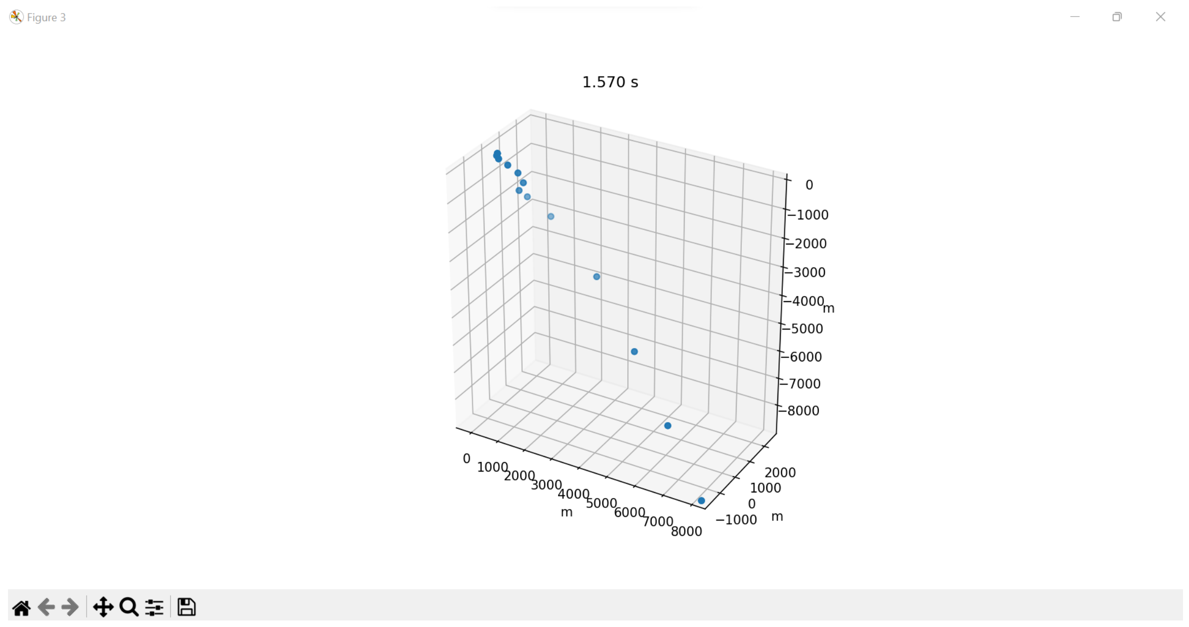


Figure 1
Sensors data, Euler angles, and AHRS internal states
Degrees/s
25000
0
−25000
Gyroscope X
Gyroscope Y
Gyroscope Z
g
25000
0
−25000
Accelerometer X
Accelerometer Y
Accelerometer Z
Degrees
100
0
−100
Roll
Pitch
Yaw
Degrees
50
0
Acceleration error
True
False
Accelerometer ignored
0.05
0.00
−0.05
Acceleration rejection timer
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
Seconds



For test 3, after creating the model, the team found that it did not match the size of the test tennis racket and recreated the work. In the end, all chips were successfully loaded into the case and held in place with tape. No detachment occurred during the test.

7 Conclusions and Recommendations for Future Work

Lessons learned during this project include hard skills for the development of the product, importance of team dynamic, and about the engineering design process. Explanation on each lesson learned follows:

1. Hard Skills on Arduino IDE/ Hardware: Lack of knowledge regarding the software we had to use was a challenge. After rounds of trails and error and with the help of YouTube tutorials and Arduino forums, we were able to work through this challenge and develop a system that worked.
2. Team Dynamic: We had trouble managing time with each member's schedules and submitting deliverables on time, therefore we had to submit some work last minute. Throughout this project, we have gotten better at managing time more effectively and set specific deadlines to make sure everybody finished their tasks. We also learned the importance of communication within the group to ensure everybody was updated on the status of the project.
3. Engineering design process: We learned how to empathize with our client and understand her needs for the product not only through her verbal cues, but also her non-verbal cues. We also learned that sometimes ideas or asking more questions to the client can help define the problem better.

If we had a few more months to work on this project, we would explore the marketing side of this project. Our client had expressed their idea of marketing our product and that is an aspect of the product we would like to explore. Additionally, we would like to improve the user interface of our website and make more tasks automated so that it can be marketed and easily available to a larger audience.

Overall, our group has turned our challenges into lessons learned through the course of this project. In the future, there are aspects of our project that we would like to improve and explore.

8 Bibliography

https://handiko.github.io/JDY-08-Reflash/

https://www.researchgate.net/figure/Connections-of-Arduino-Nano-board-to-MPU6050-module-and-laptop_fig4_353674719

APPENDICES

9 APPENDIX I: Design Files

Table 3. Referenced Documents

Document Name	Document Location and/or URL	Issuance Date
Deliverable A	Deliverable A Link	2022/12/11
Deliverable B	Deliverable B Link	2022/12/11
Deliverable C	Deliverable C Link	2022/12/11
Deliverable D	Deliverable D Link	2022/12/11
Deliverable E	Deliverable E Link	2022/12/11
Deliverable F	Deliverable F Link	2022/12/11
Deliverable G	Deliverable G Link	2022/12/11
Deliverable H	https://makerepo.com/Vivethen/1383.perrackformance-gng2101-group-b12-fall-2022 → Download deliverable from listed documents in link.	2022/12/11
Deliverable J	https://makerepo.com/Vivethen/1383.perrackformance-gng2101-group-b12-fall-2022 → Download deliverable from listed documents in link.	2022/12/11

MakerRepo Link	https://makerepo.com/Vivethen/1383.perrackformance-gng2101-group-b12-fall-2022	2022/12/11

10 APPENDIX II: Other Appendices

You can include other critical and important work here. Maybe they are not important in the structure of this document but need to be included.

Receiving arduino code:

```
#include <Wire.h>



// Serial data variables ------------------------------------------------------

//Incoming Serial Data Array

const byte kNumberOfChannelsFromExcel = 7;

float t,;

float data[6];


// Comma delimiter to separate consecutive data if using more than 1 sensor

const char kDelimiter = ',';

// Interval between serial writes

const int kSerialInterval = 50;

// Timestamp to track serial interval

unsigned long serialPreviousTime;
```

```
char* arr[kNumberOfChannelsFromExcel];


// SETUP ---------------------------------------------------------------------
void setup() {
  // Initialize Serial Communication
  Serial.begin(9600);
}


// START OF MAIN LOOP --------------------------------------------------------
void loop()
{
  // Gather and process sensor data
  processSensors();

  // Read Excel variables from serial port (Data Streamer)
  processIncomingSerial();

  // Process and send data to Excel via serial port (Data Streamer)
  processOutgoingSerial();
  }



// SENSOR INPUT CODE---------------------------------------------------------
void processSensors()
{
  t+=0.02;



  // Read sensor  and store to a variable
  data =serial.read();

  }

// Add any specialized methods and processing code below



// OUTGOING SERIAL DATA PROCESSING CODE----------------------------------------
void sendDataToSerial()
{
  // Send data out separated by a comma (kDelimiter)

for (int i = 0; i <6; i++) {

   Serial.println(data[i]);
   Serial.println(kDelimiter);





  Serial.println(); // Add final line ending character only once
}

//----------------------------------------------------------------------------

// OUTGOING SERIAL DATA PROCESSING CODE----------------------------------------
void processOutgoingSerial()
{
   // Enter into this only when serial interval has elapsed
  if((millis() - serialPreviousTime) > kSerialInterval)
  {
    // Reset serial interval timestamp
    serialPreviousTime = millis();
    sendDataToSerial();
  }
}

// INCOMING SERIAL DATA PROCESSING CODE----------------------------------------
void processIncomingSerial()
{
  if(Serial.available()){
    parseData(GetSerialData());
  }
}

// Gathers bytes from serial port to build inputString
char* GetSerialData()
{
  static char inputString[64]; // Create a char array to store incoming data
  memset(inputString, 0, sizeof(inputString)); // Clear the memory from a pervious reading
  while (Serial.available()){
    Serial.readBytesUntil('\n', inputString, 64); //Read every byte in Serial buffer until line end or 64 bytes
  }
  return inputString;
}

// Seperate the data at each delimeter
void parseData(char data[])
```

```
{
   char *token = strtok(data, ","); // Find the first delimeter and return the token before it
   int index = 0; // Index to track storage in the array
   while (token != NULL){ // Char* strings terminate w/ a Null character. We'll keep running the command until we hit it
     arr[index] = token; // Assign the token to an array
     token = strtok(NULL, ","); // Conintue to the next delimeter
     index++; // incremenet index to store next value
   }
}
```

Sending arduino code:

```
#include <ArduinoBLE.h>
#include <Arduino_LSM9DS1.h>
#include <iostream>
#include <string>
#include <sstream>

//----------------------------------------------------------------------------------------------------------------------
// BLE UUIDs
//----------------------------------------------------------------------------------------------------------------------
#define BLE_UUID_FILE_NAME                "8B37FBAC-5E52-172E-B045-1D89CCF674D1"
#define BLE_UUID_TEST_SERVICE             "9A48ECBA-2E92-082F-C079-9E75AAE428B1"
#define BLE_UUID_COUNTER                  "1A3AC130-31EE-758A-BC50-54A61958EF81"
#define BLE_UUID_RESET_COUNTER            "FE4E19FF-B132-0099-5E94-3FFB2CF07940"

//----------------------------------------------------------------------------------------------------------------------
// BLE
//----------------------------------------------------------------------------------------------------------------------
```

```
BLEService testService( BLE_UUID_TEST_SERVICE );

BLEStringCharacteristic fileNameCharacteristic( BLE_UUID_FILE_NAME, BLERead | BLEWrite, 100 );

BLEUnsignedLongCharacteristic counterCharacteristic( BLE_UUID_COUNTER, BLERead | BLENotify );

BLEBoolCharacteristic resetCounterCharacteristic( BLE_UUID_RESET_COUNTER, BLEWriteWithoutResponse );



String fileName = "";


void setup()

{

  Serial.begin( 9600 );

  //  while ( !Serial );

  while ( !Serial );


  BLE.begin();


  // set advertised local name and service UUID:

  BLE.setDeviceName( "Arduino Nano 33 BLE" );

  BLE.setLocalName( "Arduino Nano 33 BLE" );

  BLE.setAdvertisedService( testService );


  // BLE add characteristics

  testService.addCharacteristic( fileNameCharacteristic );


  // add service

  BLE.addService( testService );


  // set the initial value for the characeristic:

  fileNameCharacteristic.writeValue( fileName );
  // start advertising
  BLE.advertise();

  if ( !IMU.begin() )
  {
    Serial.println( "Failed to initialize IMU!" );
    while ( 1 );
  }
  Serial.print( "Accelerometer sample rate = " );
  Serial.print( IMU.accelerationSampleRate() );
  Serial.println( " Hz" );


} // setup


void loop()
{
  static unsigned long counter = 0;
  static long previousMillis = 0;
  long accelerationX, accelerationY,accelerationZ;
  // listen for BLE peripherals to connect:
  BLEDevice central = BLE.central();

  if ( central )
  {
    Serial.print( "Connected to central: " );
    Serial.println( central.address() );
```

```
while ( central.connected() )
{ {
   counter = 0;
  }

  long interval = 2000;
  unsigned long currentMillis = millis();
  if( currentMillis - previousMillis > interval )
  {
   previousMillis = currentMillis;

   Serial.print( "Central RSSI: " );
   Serial.println( central.rssi() );

   if( central.rssi() != 0 )
   {
    float accelerationX, accelerationY, accelerationZ;
    if ( IMU.accelerationAvailable() )
    {
     IMU.readAcceleration( accelerationX, accelerationY, accelerationZ );
    }
    float GyroX, GyroY, GyroZ;
    if ( IMU.accelerationAvailable() )
    {
     IMU.readAcceleration( GyroX, GyroY, GyroZ );
    }

    counter++;
```

```
          counterCharacteristic.writeValue( counter );
        }
      } // intervall
      if ( 1==1 )
      { String accelX = String(accelerationX, 3);
        String accelY = String(accelerationY, 3);
        String accelZ = String(accelerationZ, 3);
        String gX = String(GyroX, 3);
        String gY = String(GyroY, 3);
        String gZ = String(GyroZ, 3);
        fileName =  accelX + accelY + accelZ + gX + gY + gZ;
        Serial.println( fileName );
      }

    } // while connected

    Serial.print( F( "Disconnected from central: " ) );
    Serial.println( central.address() );
  } // if central
} // loop
```