SYSC 4907: Engineering Project

Final Report

Wearable IMU Datalogger

Supervisor: Professor Adrian D. C. Chan
Due April 14th, 2021

Table of Contents

Table of Figures

1.0 Motivation

In the field of sports analytics and research, information about athletes' movement is used to evaluate performance and prevent injury. This information is gathered using sensors that provide accurate and precise data, providing trainers and research teams with further insights into the effectiveness and safety of techniques [1,2]. Vision-based motion capture systems[1], such as the optoelectronic measurement system[2] can effectively record required data in laboratory settings [3,4], but to use this type of system in real-life requires a significant amount of time, effort, and equipment to set up [3]. Inertial Measurement Units (IMUs) on the other hand, allow the collection of precise physical motion data with minimal setup and in almost any environment [1,2,5]. This advantage has led to the rapid expansion of IMUs in sports analytics, evaluation, and improvement [1,2,5].

Numerous products now exist to utilize this movement information in the context of improving sports practices with their data, employing wearable sensors- such as on armbands, clips, and tucked under swim caps- or ones attached to sporting equipment [1,5]. These products offer a range of features, but there are some gaps in the market, as outlined in the feature comparison in Table 1:

Table 1: Wearable Sensor Market Analysis

	TritonWear Triton 2 [9]	*MbientLabs* MetamotionC [8]	I Measure U/Blue Trident [7]
Modularity/ Customization options	None	Able to integrate any additional sensor, but not additional hardware such as storage	Partial, can be integrated with other sensors from the same system

[1] Sight-based sensors are visual based sensors that use cameras to determine orientation and motion of an object.

[2] Optoelectronic measurement system[2], is a system that uses optoelectronic cameras to detect light and estimate 3D orientation of an object.

Included sensors	9-DoF[3] IMU	9DoF IMU, Barometric Pressure, Temperature, Ambient light	9-DoF IMU, high-g accelerometer[4]
Accelerometer range	(Not available on product site)	+-2 ,+-4, +-8, +-16g	Low-g ±16g / High-g ±200g
Gyroscope range	(Not available on product site)	± 125, ± 250, ± 500, ± 1000, ± 2000°/s	± 2000°/s
Data Storage	None (wireless data transmission only)	8MB	1GB
Wireless Connection	RF (IMU to receiver) Bluetooth (receiver to pc/tablet)	BLE only (decreases maximum sampling rate)	Bluetooth 5
price	Hardware comes with annual software subscription (188 USD/year)	81.99 (75.99 for pcb-only version)	1,600.00 USD

Other systems, including the *Phlex* Edge and *Moov* Now systems were also researched, but as these products are designed for individual athletes and other non-technical audiences, little technical information is available about the sensors themselves and they do not appear to allow for the extraction of raw sensor data or even direct access to orientation data. As can be observed from the feature comparison table, one notable shortcoming is that consumer sensor systems tend not to be modular. While a practical decision for a non-technical audience that simply wants the post-processing results, in the context of research, this means that these systems lose the ability to integrate additional sensors data and only provide movement data generated from the IMU. The out-of-the box convenience also tends to come at the expense of adequate storage for longer testing (or the ability to add it) even in systems designed to provide top-of-the-line analytics, severely limiting their effectiveness in use cases where testing occurs over a longer duration outside of wireless range. In the research context that this project is aimed at, these

[3] Degrees of freedom, number of logically independent values measured.
[4] Accelerometer, measures acceleration forces on an object.

features can be used to provide a more complete view of the athlete's situation during testing and will have the potential to make the wearable sensor system more adaptable for researching different sports [5]. This sort of modular system will create a market that will allow for future sensors to become standardized, they can easily integrate with this system. In this way, users can create customized systems to fit their needs. Future sensors could provide environmental information (temperature sensors, barometers, or even light sensors), biomedical information (integrated vitals monitoring), or physical and timing information (force sensors/switches) in addition to the movement data from the IMU.

It is also noted that despite the advancements that have been made, a select group of athletes have been neglected the benefits of this potential assessment and coaching. In 2018, a systemic review of the trends supporting the use of wearable inertial sensors noted that only five papers out of potential two thousand (0.002% of all papers) incorporated para-athletes [1]. To attempt to address this disparity, further research and testing is required is required for para-athletes. The focus of this project is the development of one such device, that can be customized enough to be appropriate for the monitoring of para-ice hockey (PIH), also known as Sledge Hockey. The idea of a modular and adaptable system is perfect in this regard, as such adaptation may be needed for to support the testing of parasport and para-athletes. Our system addresses the main shortcomings caused by commercial devices' lack of modularity: custom sensor input- to improve the device's use in a research context- and storage space- as sledge hockey testing may require longer testing durations and a test performed on a standard ice rink will not always be within wireless range. It is also designed with being appropriate for use at sub-freezing temperatures, in mind and is reasonable to mount on the specialized equipment for the sport.

2.0 Objective and Goals

The short-term objective of this project is to design, implement, and test a wearable prototype that can reliably collect and store data from an IMU for the assessment and improvement of sports performance. This system will be able to transmit and receive data wirelessly, as well as output real-time data. In addition to this, the system can be fitted with complementary sensors to adapt to different activities and sports as needed. For example, heart rate, GPS, and force sensors can be added to provide as much information as needed to the user for sports analysis.

The overall long-term objective of this project is to develop a system model which incorporates the use of sledge hockey analytics to improve the concerned athletes’ performance while minimizing their risk of injury in the process. This in turn would have a profound impact on the overall functioning of the team. In addition, because of how the system is designed, it can be easily adapted to other sporting activities ex. table tennis, football, or volleyball among others, to provide the same performance increase and injury reduction.

To achieve the short-term objective, there are short term goals set to aid in the elaboration of this system. These short-term goals are:

1) Designing a wearable system that records data from an IMU and other sensors.
2) Enable the system to stream real-time data wirelessly to a computer.
3) Describe the motion data and attempt to quantify what the subject is doing.
4) Demonstrate that the data can be processed and displayed on a computer.

3.0 Scope

The general purpose of the project is to provide low-cost, modular, and simple sports analysis to all who require it, that will fit their needs. To achieve this the project must be limited to meet these needs over the duration of September 2020 to April 2021.

Low-Cost: The base system (without additional sensors) should cost no more than one hundred dollars.

System: Analyzing the movement data in relation to performance for specific sports is out of scope. The prototype system will be able to record data, transmit the data to a computer, describe the movement of a subject, and load the data to a high-level language for processing.

Low-Fidelity Prototype: To build a high-fidelity prototype is also beyond the scope of this project. The prototype under development will be made from existing microcontroller and breakout boards and focus on basic functionality rather than a consumer-grade product. Issues such as form factor and case are out of scope, but the final project is soldered to a permanent proto board.

Real-Time Display: While there could eventually be a goal of real-time analysis, the scope of this project is limited to real-time display (for calibration) and offline analysis, surface level analysis can be done to the real-time data for minor observations.

4.0 Proposed Solution

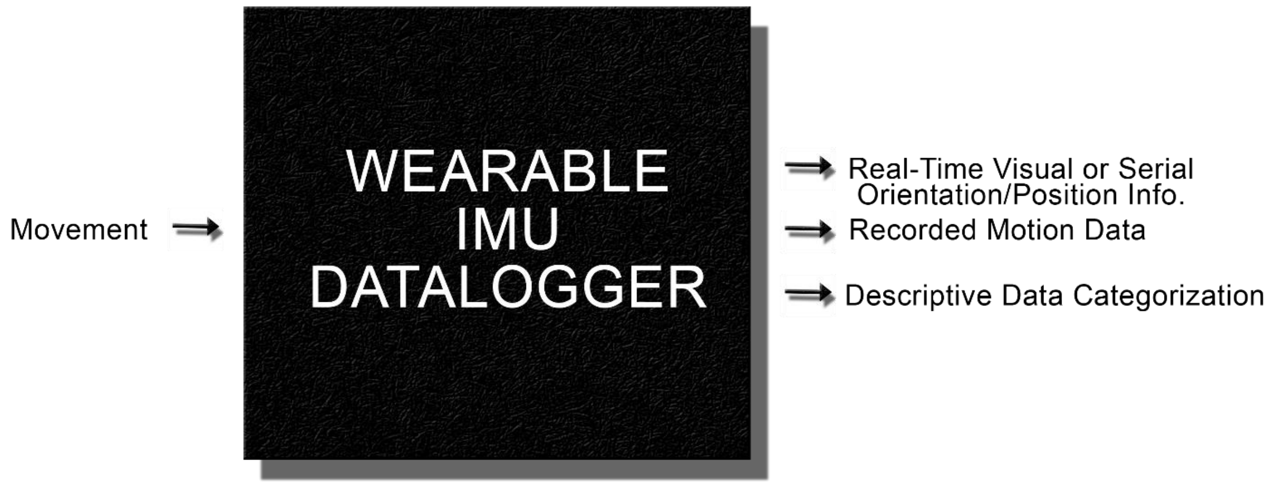


Figure 1: Wearable IMU Datalogger Black Box

Referring to Figure 1, our proposed wearable sensor system will take movement input and produce real-time visual or serial orientation data, recorded motion data, and describe the movement of an intended subject. The real-time visual or serial orientation information can be used to perform surface level analysis on the system. The recorded motion data is where most of the analysis will be performed, after recording an intended workout or sports activity, the data from the activity can be analyzed with software to make assessments to improve one's sports performance. Finally in addition to the recorded data, our system will attempt to describe the athlete's movement and append the description to the recorded motion data and output it live. In conclusion, our wearable sensor system will facilitate the transformation of motion data into clear visual data, described recorded motion data, and used to analyze performance shortfalls or injury risks.

4.1 High Level Overview

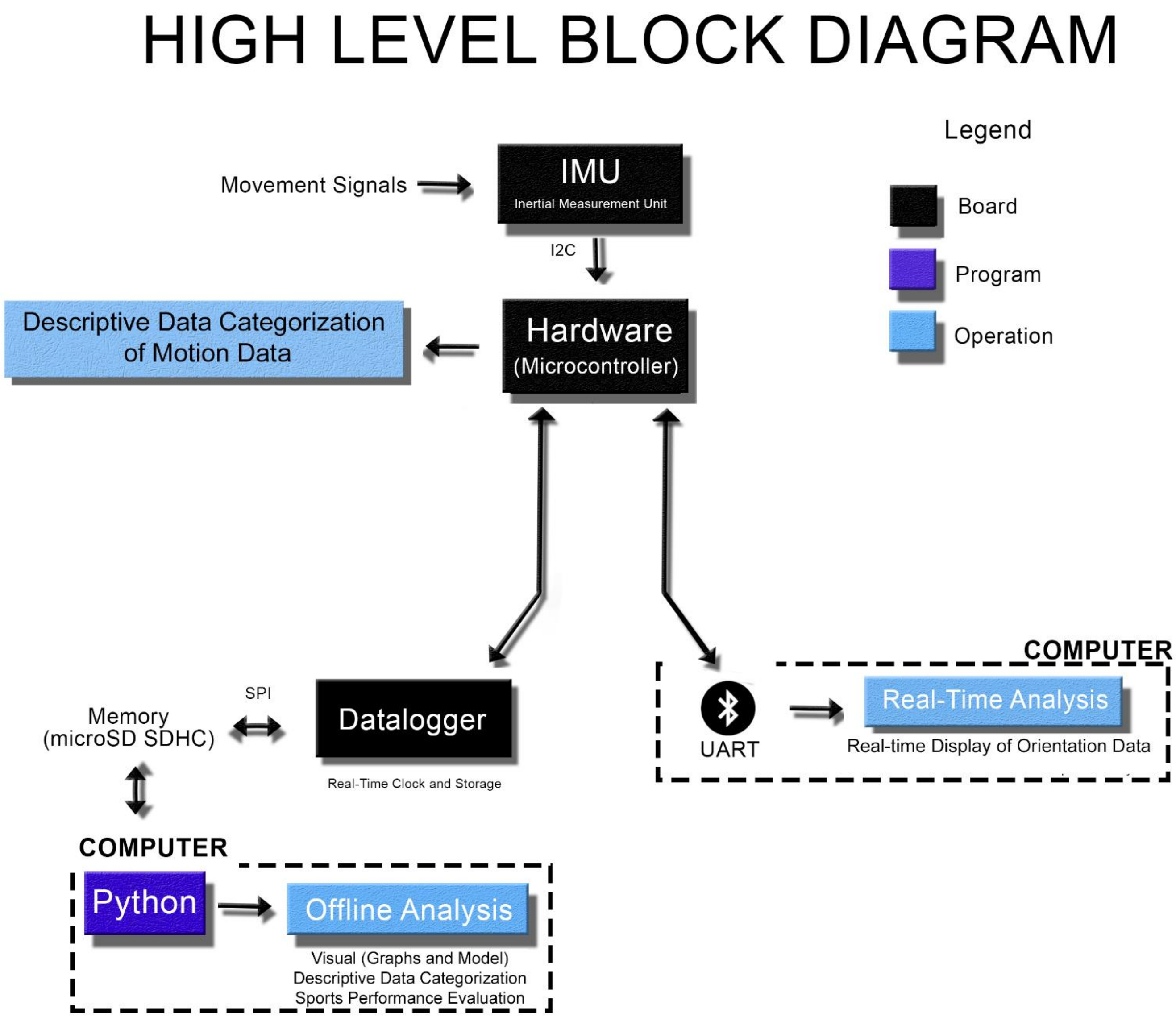


Figure 2: Wearable IMU Datalogger High-Level Diagram

Figure 2 describes the proposed system that consists of sensors, a microcontroller, and a computer for real-time display, offline display, and processing. The system will have an IMU that collects accelerometer, gyroscope[5], and magnetometer data, as well as additional sensors (heart rate, GPS, and force sensors) that can be added via communication protocols SPI[6] and I2C[7] or an analog signal that is converted by the microcontrollers analog-to-digital converter. Sensor data is received by the microcontroller, which can store it to non-volatile memory, in the form of an SD card, or transmit it over Bluetooth for real-time display. A new addition since the project proposal is the determination of operation modes.

The system will have two operation modes:

1) Bluetooth mode, where it transmits data over Bluetooth for real-time output to validate that the device is measuring accurate data before recording. Data is transmitted for quick viewing and shallow analysis.
2) Datalogging mode, in which the system writes the IMU sensor data to the SD card for offline analysis for obtaining stable recorded data when the subject might leave wireless range. This offline analysis is where the data will be used to gauge an athlete's performance metrics via chosen tool of the end-user.

At the high-level, our proposed system will take the movement from the user equipped with the device, then output calibrated and accurate orientation and position characteristics.

4.2 Design Requirements

To achieve the desired solution, the proposed system must meet minimum requirements that can include power, ranges, and operating factors to perform the objective satisfactory. To

[5] Gyroscope, measures angular velocity in degrees/sec.
[6] Serial Peripheral Interface, interface bus used to send data between the microcontroller and additional peripherals (fast, usually reserved for SD Cards).
[7] Inter-Integrated Circuit, communication protocol that allows for up to 128 to communicate on the same interface.

determine what components are necessary for the project, these minimum requirements were essential to determine component choices for the system.

4.2.1 Sensors

The system must include a 6DoF IMU sensor at minimum to provide adequately accurate information for analysis. A 9DoF IMU provides users with even more potentially useful information but is not required. A minimum range of ± 3g of acceleration and 450 deg/s of rotation is required to meet the basic function of our project.

4.2.3 Microcontroller

The microcontroller should include enough channels to support the IMU sensor, SD Card, and additional sensors if required by the user. For this purpose, at least one SPI channel and five I2C channels are needed from the microcontroller. The speed of the channels must be adequate to support measure the forces, movement, and directional changes expected from the end-user during their sporting activities. A bus speed of 50 MHz is required for the SPI channel, and at least 0.1 kbps for the I2C channels.

4.2.4 Storage

The system must be able to collect and store at least five hours of continuous data from all sensors at an acceptable sampling rate. An acceptable sampling rate in this case is defined by having recorded enough data so that it viable in the analysis of activity by the end-user. Based on previous testing, a sampling rate of approximately 50Hz would be adequate for analysis [4]. To sample only accelerometer and gyroscope data at this rate for the established minimum testing time would require approximately 5MB of memory, but as this system is designed to be modular, it should have the ability to add more storage (i.e., it should have a user-accessible SD card slot). The data recorded must be non-volatile and have a date associated with it.

4.2.5 Wireless

While the system should be designed to be able to operate within of wireless range, there are some requirements of the wireless functionality. Firstly, the system should be able to perform short-distance transmission to a computer. The operating range should be within ten to twenty meters, and easily connect to devices. The wireless communication protocol must be widely available on PCs and devices (secluded frequency range). The bandwidth this data is sent in must not conflict or compete with other wireless devices. Also, the data transmitted needs to be at least transmitted at 2 kbps.

4.2.6 Size

The device must be an appropriate size to be used in sledge hockey trials or as a general wearable device for various sports. Based on previous experiments, a device used for sledge hockey should be small enough to be mounted at the base and backrest of a sledge [4] as shown in Figure 3 below, where the sensor is represented by the purple rectangular block placed below the sled or behind the backrest. To make the device useable in these trials and non-obtrusive for other sports, the target size was determined around under 3.3 x 2.4 x 0.6 inches or smaller (a size that is comfortable enough to worn on the body for a long period of time). When not used for sledge hockey, the system can be strapped to the back of the wearer as seen again in Figure 3.

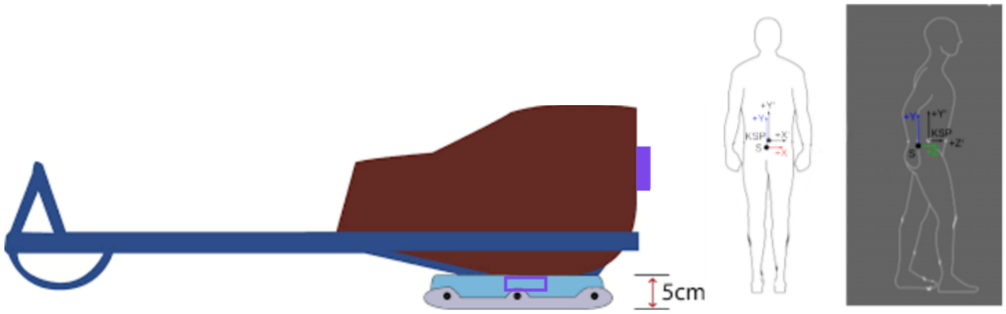


Figure 3: Sensor Mounting Locations [6]

4.2.7 Battery

The entire system must be powered by a source that can maintain the integrity of all components. To be appropriate for the longer-duration trials it seeks to accommodate, the system must remain powered on for a duration of a five hours and preferably be recharged with a power source separate from the mounted datalogger casing. A battery of at least 500 mA is required to power the basic system and additional sensors for the expected five-hour use case.

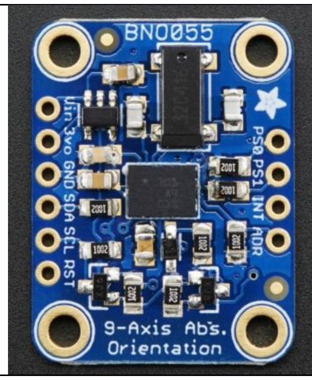
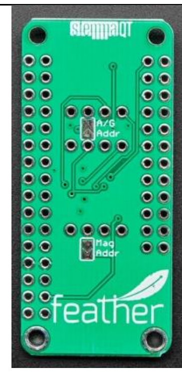
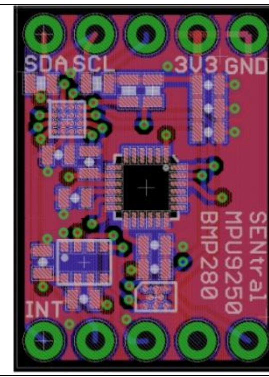
4.3 Design Choices

As this project is concerned with the development of a prototype, the use of a development board is the most reasonable choice as it easily allows for the full implementation of multiple versions (corresponding with individual attempts or milestones). With the minimum requirements laid out to achieve the objective, component recommendations were made, and parts were purchased for the proposed solution. This section explains the primary options that were investigated and the rationale behind the final choices that were made with regards to the hardware components chosen.

4.3.1 Inertial Measurement Unit (IMU)

Table 2: IMU Option Comparison

	Adafruit BNO055	Adafruit FeatherWing	MPU9250 - Ultimate Sensor

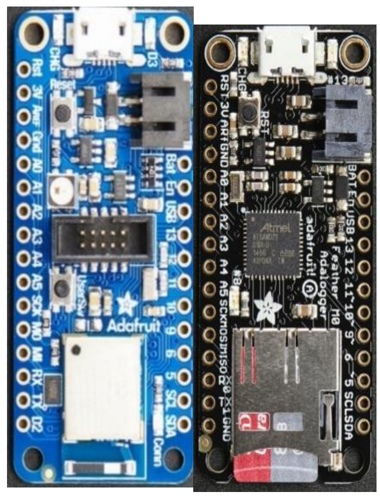
		LSM6DSOX + LIS3MDL	
			
Fusion Algorithm	Built-in	User controlled	Built-in
Adequate Operation Range	**Gyroscope** rotational speed in °/s 125-2000 °/s @100Hz **Accelerometer** Gravity and Linear Motion in m/s ±2 to ±16 g @ 100Hz **Magnetometer** Magnetic Field in µTesla ±1200-1300 µT in x-y and ±2000-2500 µT @ 20Hz	**Gyroscope** rotational speed in °/s 125-2000 °/s @100Hz **Accelerometer** Gravity and Linear Motion in m/s ±2 to ±16 g @ 100Hz **Magnetometer** Magnetic Field in µTesla ±1200-1300 µT in x-y and ±2000-2500 µT @ 20Hz	**Gyroscope** rotational speed in °/s 125-2000 °/s @100Hz **Accelerometer** Gravity and Linear Motion in m/s ±2 to ±16 g @ 100Hz
Additional Sensors	Temperature	None	None
Size	0.15 x 0.2 inches	2.0 x 0.9 inches	0.5 x 0.7 inches
Environmental Stress	Good	Good	Cannot handle direct sunlight
Power Management	Deep Sleep, Low-Power, and Suspension Mode	None	No power saving features, 1% power usage compared to ARM Cortex
Bonus Features	Dual Mode (Raw or fused movement data)	None	64 kB EEPROM to store calibration data (for warm start)
Peak Power Draw	12.3 mA	25 mA	14.8 mA
Price (US)	$34.95	$14.95	$35.95
Interface	I2C	I2C	I2C
Transmission Speed	3.4 Mbps	3.4 Mbps	3.4 Mbps

As shown in the table, All the sensors listed meet the minimum required axes measurement (all meet the minimum six degrees of freedom). There are other options with less drift and inaccuracies, however, the difference was negligible in comparison to the extra power, size, and cost associated with them. Each IMU listed has acceptable peak power draw for the purposes of our system (under 30 mA), but of the options listed the MPU9250 has the lowest power usage. The MPU9250 and BNO055 both have built in fusion algorithms, however, the MPU9250 is incapable of outputting raw acceleration, gyroscope, and magnetometer data. The FeatherWing LSM6DSOX + LIS3MDL is better in this regard because the raw data can be integrated to suit any environment surpassing the usefulness of a built-in data fusion. This is important, as the built-in fusion algorithm could be deemed inadequate for a situation, leaving the end-user unable to adjust the data for better results. The BNO055 and the FeatherWing LSM6DSOX + LIS3MDL can handle wet, humid, warm, bright conditions whereas the MPU9250 cannot. For a wearable system, being adaptable to different conditions is vital, especially given that this system specifically intends to be usable under sporting conditions that have the potential to damage the MPU9250. For this reason, and the inability to access raw sensor data, the MPU9250 is not a viable choice. As a bonus feature the BNO055 includes a temperature sensor. All three sensors communicate with the microcontroller via I2C, since we are sampling at a max of 100Hz, the I2C transfer speed of 3.4Mbps is more than adequate for our purposes. The BNO055 better suited for multiple environments, acceptable power usage, and both fused and raw motion data. Additionally, the BNO055 does this all whilst having a smaller size, imprint, and power draw. For these reasons, we chose BNO055 as it was the best option, but did cost more than the FeatherWing.

4.3.2 Microprocessor and Addons

Table 3: Microprocessor/Addon Option Comparison

	Adafruit nRF52840 Express + Adafruit Feather M0 Wi-Fi	Adafruit HUZZAH32 + Adalogger FeatherWing

		
Bluetooth	✓ (BLE only)	✓ (BLE & Classic)
Wi-Fi	✓	✓
Included Datalogging	✓	✓
Flash	1 MB/256 KB	4 MB
SRAM	256 KB/32KB	520 KB
Size	2.0 x 0.9 inches/2.0 x 0.9 inches (will be stacked)	2.0 x 0.9 inches (will be stacked)
Real-Time Clock	X	✓
Dedicated SPI/UART/I2C Channels	6 SERCOMS that are capable of SPI/UART/I2C	3 x SPI
SPI Transfer Speed	50 MHz	50MHz
Dedicated UART Channels	6 SERCOMS that are capable of SPI/UART/I2C	3 x UART
Dedicated I2C Channels	6 SERCOMS that are capable of SPI/UART/I2C	2 x I2C
Transfer Speed	3.4 Mbps	3.4 Mbps
Portable Battery	✓	✓
Processor	Cortex-M4 processor	dual-core ESP32

For the purposes of creating a robust prototype, we are looking for an Arduino compatible microcontroller for ease of development. The first option of an Adafruit nRF52840

Express and Adafruit Feather M0 Wi-Fi most notably has support for the Adafruit Bluefruit application, this application would allow for real-time calibration of our system via iOS or Android devices. However, this option has much lower flash memory for Arduino sketches to be uploaded, as well as reduced variable manipulation speed due to the lower SRAM. The dual-core ESP32 processor is also faster than the Cortex-M4, with an extra core to boot. Both systems meet the required specifications for the proposed system, however the HUZZAH32 and Adalogger performs better in several areas, including the previously mentioned higher specifications, more dedicated channels, and Wi-Fi capability, as well as being a singular entity to upload code to (as the first option requires sketches to be uploaded to both devices). The ESP32 is also the more common system for wireless projects, universally supported, and more compatible for additional sensors/devices. For these reasons, the combination of Adafruit HUZZAH32 and Adalogger FeatherWing was chosen as they are more versatile and provide more processing power and features than required. For the goal of the project, the Adafruit HUZZAH32 board was chosen as it meets and exceeds all identified hardware and form factor requirements.

4.4 Low-Level System Overview

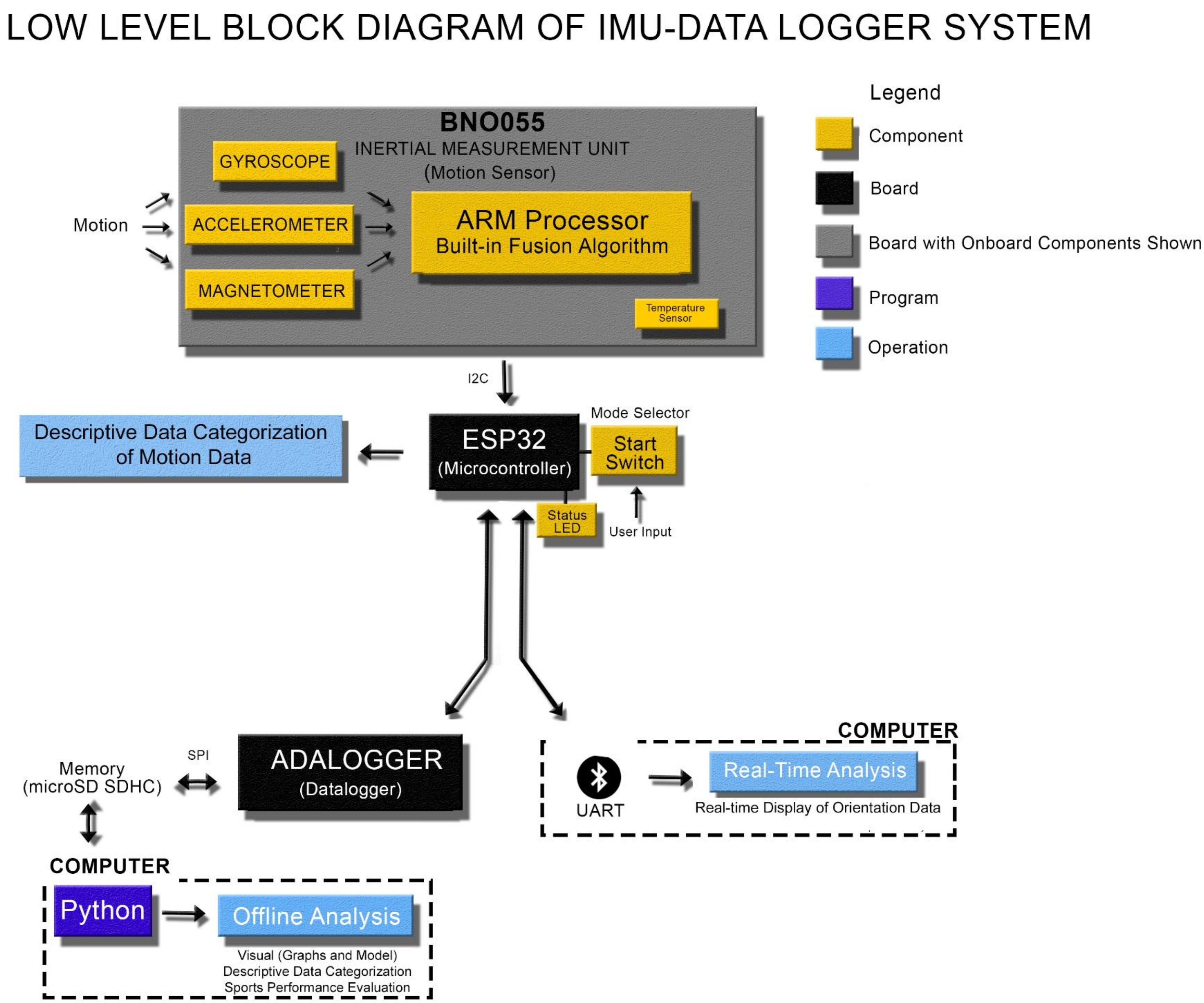


Figure 4: Wearable IMU Datalogger Low-Level Diagram

As seen on Figure 4, using the BNO055 as our IMU, the system will collect rotational speed in degrees per second, acceleration due to gravity in m/s, the magnetic field strength in µTesla, and fuse that data into usable orientation data via the BNO055's built in Cortex-M0 processor. The BNO055 has a built-in fusion algorithm that combines the data from an accelerometer, gyroscope, and magnetometer into useable orientation position, and quaternion data. The BNO055 provides a temperature output that is also sampled and processed by our system. Using the standard I2C communication protocol the data is sent to the HUZZAH32 microcontroller for processing and output to the PC or Adalogger as determined by the operating mode. After the orientation data is sent to the microcontroller, it is compared to the previously sampled orientation data to determine the system/subject's movement. Depending on the change in orientation, the HUZZAH32 will output what it believes is happening. The orientation and temperature data are sent to the SD Card via the Adalogger and SPI. Once received by the Adalogger quaternions, temperature, calibration, date and described movement are saved to the SD Card every second, as well as shown outputted on serial. This is the default behaviour of our system, however the operating mods can be changed with a push button and shown via a status LED. By default, datalogging is represented with a lit status LED, when the push button is pressed the system switches to Bluetooth transmission mode and the status LED is unlit. In this operation mode, live orientation data is sent to the PC, and the data stream can be halted by commands from the PC. Overall, this system will operate like this until the battery dies, and when powered on again will retain the time because of the I2C real time clock with battery backup the Adalogger provides.

5.0 Implementation

Implementation of the project was done in three stages to ensure that each group member had working components and to ease the development. Stage 1 was the IMU integration where the BNO055 and the HUZZAH32 were connected and tested. Stage 2 was the Adalogger Integration, where we confirmed that real-time clock and SD card could be used and recorded to. Stage 3 was Bluetooth transmission, where we took IMU data and transmitted via BLE to an appropriate PC.

5.1 Stage 1: IMU Integration

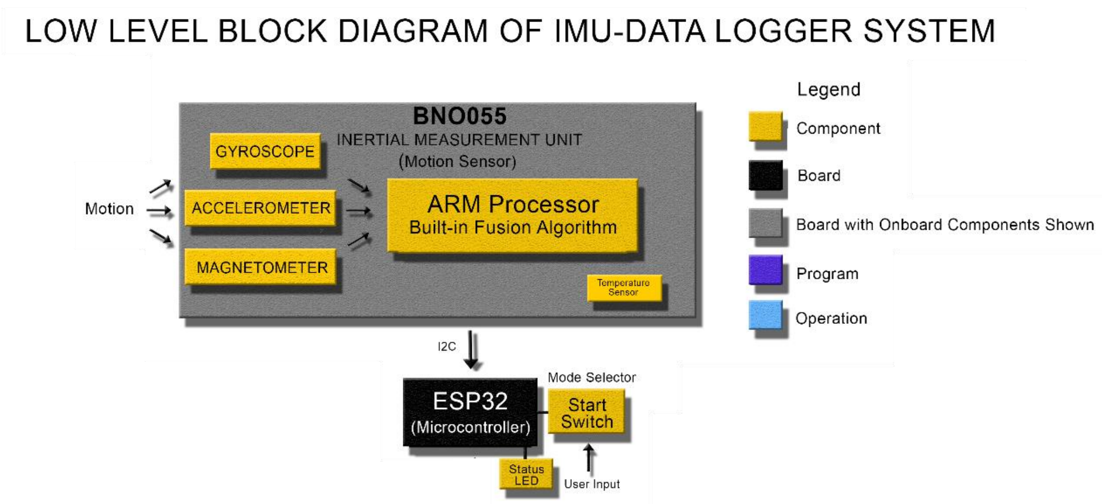


Figure 5: Wearable IMU Datalogger Stage 1 Diagram

To begin, the BNO055 IMU was integrated to the microcontroller as shown in Figure 4 and 5. To confirm that the components worked and worked together, Adafruit's web serial visualizer was used to take the IMU data and create a Stanford bunny 3D model. Once it was confirmed that the hardware connections were correct and the sensor was outputting values as expected, the sketch was modified to output sensor values, so that the system would be able to store and eventually process these values. Raw acceleration was outputted via a simple sketch, the acceleration was compared to known values such as using the acceleration due to gravity. We would place the device perpendicular to a level surface, if the value was not 9.8 m/s^2, we could not ensure the accuracy of the data being outputted. Each part of the BNO055 (accelerometer, gyroscope, and magnetometer) requires calibration, without it the fused movement data can have errors. Each part of the BNO055 will have calibration ranges from zero to three, three being full calibration and zero being uncalibrated. Ensuring that the BNO055 is calibrated is important in ensuring we get the most out of the device, as inconsistencies add up overtime. The entire BNO055 has a calibration range as well, and the full IMU calibration is based on the individual component calibration. The BNO055 works generally well at a calibration of one or two, but the most accurate data is produced when fully calibrated at three. Several sketches were used to validate the BNO055 for the requirements we set out. Quaternions, Euler angles, and orientation were outputted at 100Hz to the microcontroller and no abnormal power draw or data loss was observed. Peak power draw for our BNO055 peaked at an approximate 9.1 mA with our testing. Test sketches were done to validate that the button and status LED worked and performed as expected. To connect both components, resistors of 1k and 330 ohms were utilized as shown in Figure 6. In conclusion the IMU, LED, and button were confirmed to be working, accurate, and acceptable for our project.

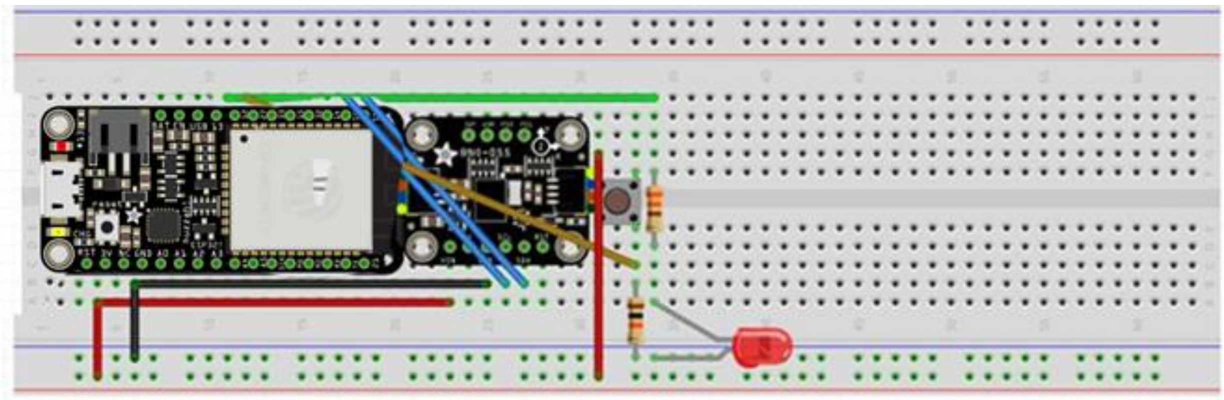


Figure 6: Wiring Diagram for Stage 1: IMU Integration [7]

5.2 Stage 2: Adalogger Integration

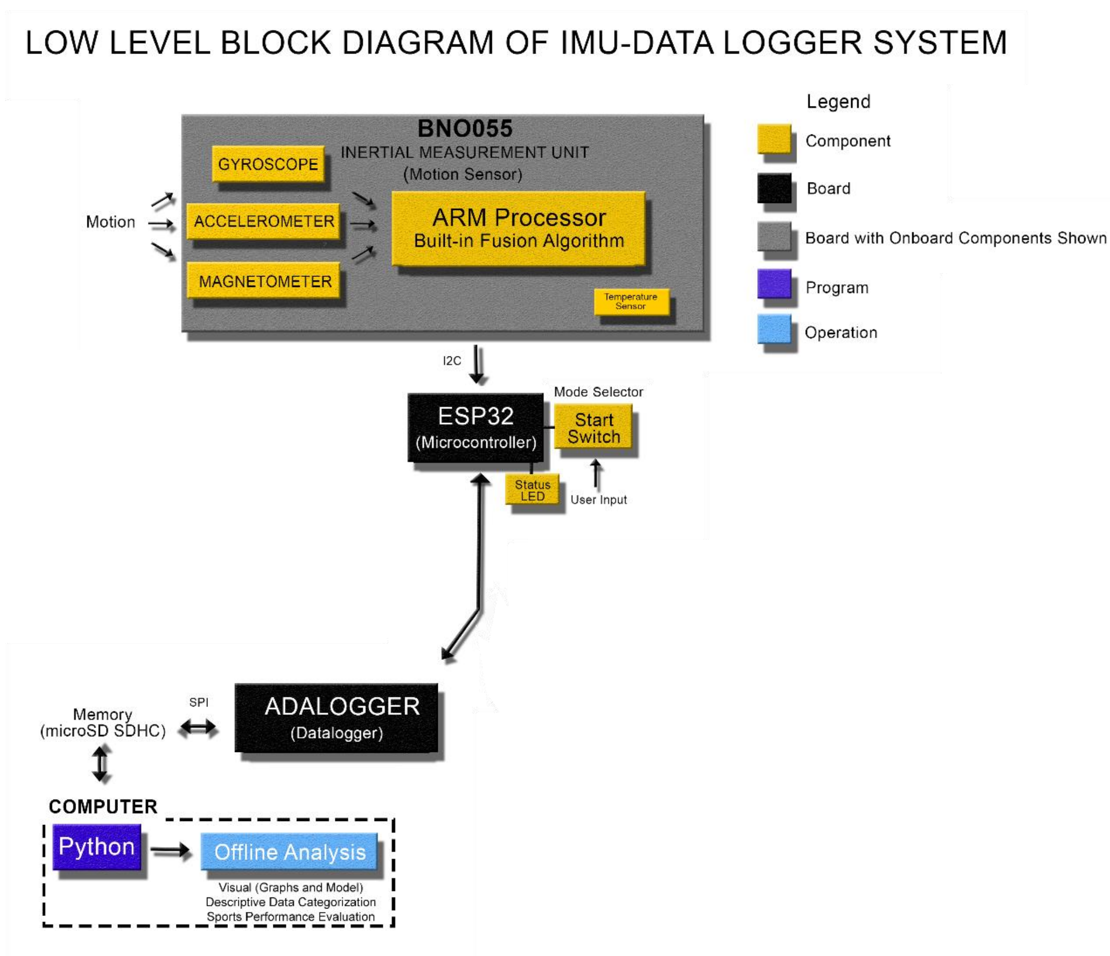


Figure 7: Wearable IMU Datalogger Stage 2 Diagram

To start, the SD Card slot was added to the device using the Adalogger FeatherWing add-on, which is designed to interface when stacked directly on top of the microcontroller as shown in Figure 8. SD Card integration was initially tested by printing known values from the microcontroller to the SD card and removing it to manually check the files. When this connection was confirmed, the sketch was modified to store IMU, time stamps, temperature, described movement, and calibration values to the SD card. Again, this was tested by removing the SD card and checking its files on a separate device for the stored data. The Adalogger also contains a I2C real time clock (PCF8523) that was tested with an example script in the RTClib.h Arduino library. With another example sketch, we were able to test the full duplex communication of the Adalogger and was able to write and read files stored on the SD card via USB. With this, the Adalogger was confirmed to be working and validated for our purposes, with this we confirm that Stage 2 was complete (Figure 7).

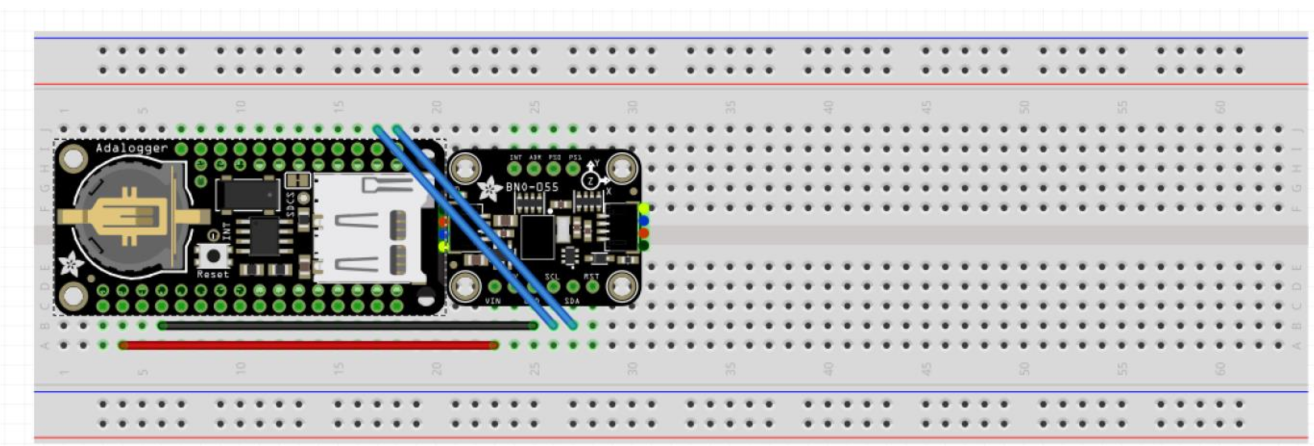


Figure 8: Wiring Diagram for SD Card Integration [7]

5.3 Stage 3: Bluetooth Transmission

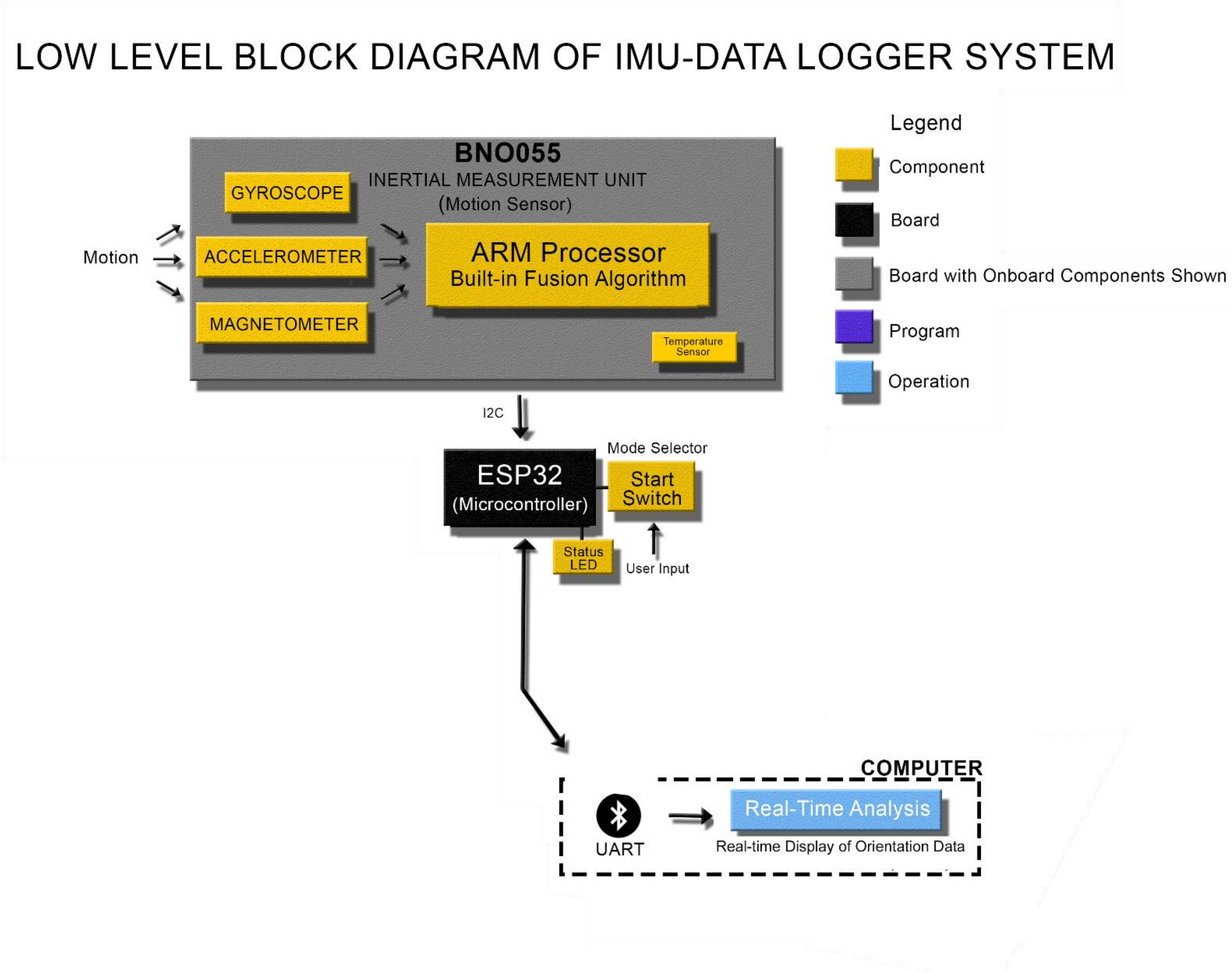


Figure 9: Wearable IMU Datalogger Stage 3 Diagram

In Stage 3, we tested the Bluetooth and Wi-Fi capabilities of the system. Wireless communication using Bluetooth Classic protocol was implemented using the Arduino Bluetooth Serial library. It was initially tested by sending strings of text from the serial monitor to the Bluetooth serial terminal. When this was confirmed, the sketch was modified to be bidirectional and send strings of text in both directions. When bidirectional communication was confirmed, the sketch was modified again to send the IMU positional and calibration values from the microcontroller to the Bluetooth serial terminal so that these values could eventually be passed through the display program. Upon confirming this functionality, the program was modified to also perform a Bluetooth serial read (at a lower frequency than the IMU data is sampled and sent, as a human is unlikely to be able to type and send commands faster than the current 10Hz IMU sampling rate) and check those serial reads for designated start and stop commands. With example sketches, a connection to the Wi-Fi access point and nearby PC's Bluetooth were established. After confirming everything was in a working order, development of sketch that would transmit orientation in x, y, z to a Bluetooth terminal began. Then the sketch detects if data is available to be read and stores it into string that sent to the PC via BLE. Once that was completed, we modified the code to accept commands from the Bluetooth terminal and to stop and start Bluetooth transmission. Finally, we adapted the code to include commands from the push button we implemented in Stage 1. This stage was complete with the testing Bluetooth capabilities, developing a sketch to output orientation data in a string wirelessly, and issuing commands to the system via Bluetooth or hardware. With this, moved onto software and full system integration.

5.4 Software Integration

To demonstrate our project's goal Arduino, Python, Serial Plotter, and Bluetooth Serial Terminal are used in the basic use and demonstration of our system. With Arduino, the sketches we upload will be used to manipulate the microcontroller for adaptive uses and operational modes. Python will be primarily used to output a 3D model depiction of our system but can also be used for the offline analysis of that same data. Serial Plotter will take the real-time orientation, position or quaternion data and plot it, and depending on the number of channels of data being outputted, this plot can be modified. Bluetooth Serial Terminal will be used to send commands to the system and to receive orientation data in strings. Using a mix of these programs and programming languages, the system will can meet all our short-term goals and objectives.

5.4.1 Arduino

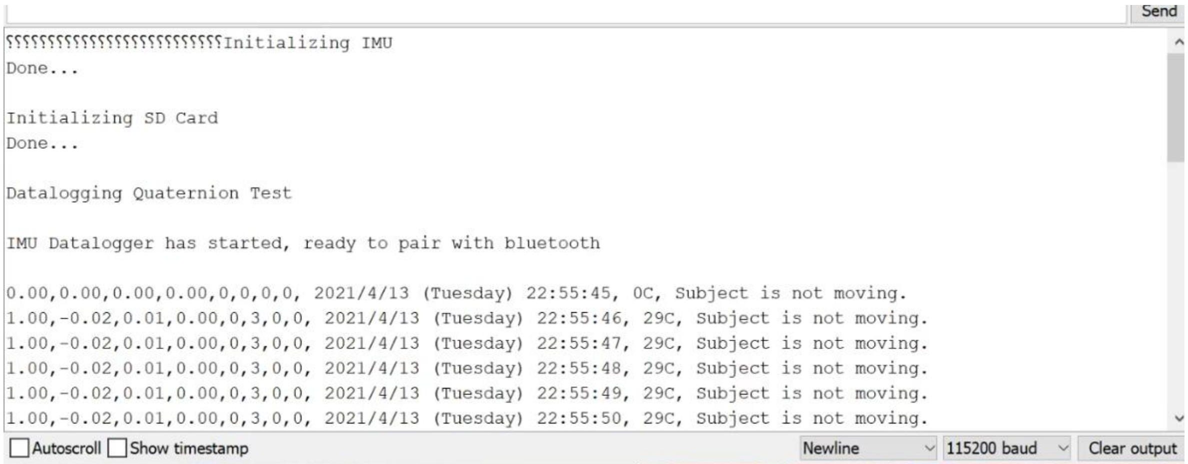
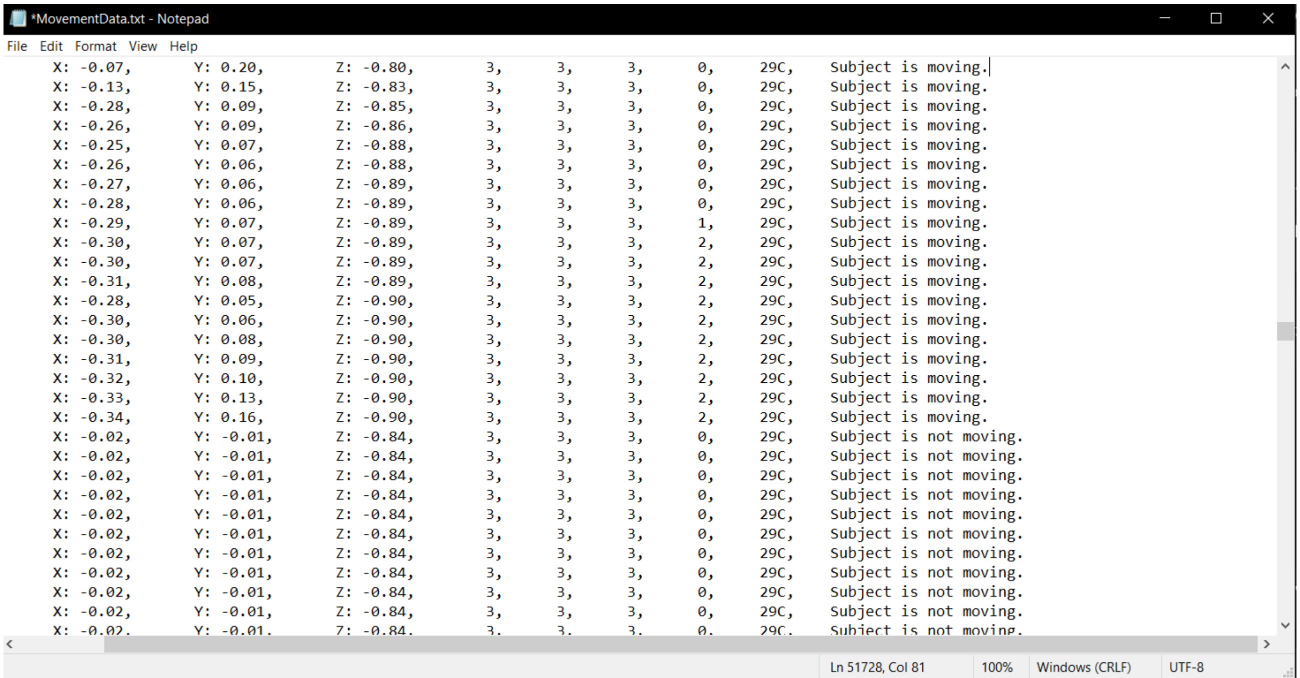


Figure 10: Out of Arduino Serial Monitor when FinalCode.ino is Run

Here in Figure 10, you can see the initialization process of our final Arduino sketch. First, the BNO055 and SD Card are initialized, if not, an error message will output indicating which one is disconnected. The device becomes available to pair with Bluetooth but will not transmit orientation data unless given the command to. Quaternions, calibration, real-time clock date, temperature, and described movement are outputted to the Arduino Serial monitor in that order, the same data is also recorded to the SD Card (Figure 11). Datalogging only occurs if switched to the datalogging operational mode with the button switch. The Arduino sketch facilitates the Bluetooth, datalogging, operation modes, and real-time clock of our system. Just as the microcontroller ties the entire project together, the Arduino code is responsible for manipulating each component.



*MovementData.txt - Notepad

File Edit Format View Help

```
X: -0.07,    Y: 0.20,     Z: -0.80,    3,    3,    3,    0,    29C,    Subject is moving.
X: -0.13,    Y: 0.15,     Z: -0.83,    3,    3,    3,    0,    29C,    Subject is moving.
X: -0.28,    Y: 0.09,     Z: -0.85,    3,    3,    3,    0,    29C,    Subject is moving.
X: -0.26,    Y: 0.09,     Z: -0.86,    3,    3,    3,    0,    29C,    Subject is moving.
X: -0.25,    Y: 0.07,     Z: -0.88,    3,    3,    3,    0,    29C,    Subject is moving.
X: -0.26,    Y: 0.06,     Z: -0.88,    3,    3,    3,    0,    29C,    Subject is moving.
X: -0.27,    Y: 0.06,     Z: -0.89,    3,    3,    3,    0,    29C,    Subject is moving.
X: -0.28,    Y: 0.06,     Z: -0.89,    3,    3,    3,    0,    29C,    Subject is moving.
X: -0.29,    Y: 0.07,     Z: -0.89,    3,    3,    3,    1,    29C,    Subject is moving.
X: -0.30,    Y: 0.07,     Z: -0.89,    3,    3,    3,    2,    29C,    Subject is moving.
X: -0.30,    Y: 0.07,     Z: -0.89,    3,    3,    3,    2,    29C,    Subject is moving.
X: -0.31,    Y: 0.08,     Z: -0.89,    3,    3,    3,    2,    29C,    Subject is moving.
X: -0.28,    Y: 0.05,     Z: -0.90,    3,    3,    3,    2,    29C,    Subject is moving.
X: -0.30,    Y: 0.06,     Z: -0.90,    3,    3,    3,    2,    29C,    Subject is moving.
X: -0.30,    Y: 0.08,     Z: -0.90,    3,    3,    3,    2,    29C,    Subject is moving.
X: -0.31,    Y: 0.09,     Z: -0.90,    3,    3,    3,    2,    29C,    Subject is moving.
X: -0.32,    Y: 0.10,     Z: -0.90,    3,    3,    3,    2,    29C,    Subject is moving.
X: -0.33,    Y: 0.13,     Z: -0.90,    3,    3,    3,    2,    29C,    Subject is moving.
X: -0.34,    Y: 0.16,     Z: -0.90,    3,    3,    3,    2,    29C,    Subject is moving.
X: -0.02,    Y: -0.01,    Z: -0.84,    3,    3,    3,    0,    29C,    Subject is not moving.
X: -0.02,    Y: -0.01,    Z: -0.84,    3,    3,    3,    0,    29C,    Subject is not moving.
X: -0.02,    Y: -0.01,    Z: -0.84,    3,    3,    3,    0,    29C,    Subject is not moving.
X: -0.02,    Y: -0.01,    Z: -0.84,    3,    3,    3,    0,    29C,    Subject is not moving.
X: -0.02,    Y: -0.01,    Z: -0.84,    3,    3,    3,    0,    29C,    Subject is not moving.
X: -0.02,    Y: -0.01,    Z: -0.84,    3,    3,    3,    0,    29C,    Subject is not moving.
X: -0.02,    Y: -0.01,    Z: -0.84,    3,    3,    3,    0,    29C,    Subject is not moving.
X: -0.02,    Y: -0.01,    Z: -0.84,    3,    3,    3,    0,    29C,    Subject is not moving.
X: -0.02,    Y: -0.01,    Z: -0.84,    3,    3,    3,    0,    29C,    Subject is not moving.
X: -0.02,    Y: -0.01,    Z: -0.84,    3,    3,    3,    0,    29C,    Subject is not moving.
X: -0.02,    Y: -0.01,    Z: -0.84,    3,    3,    3,    0,    29C,    Subject is not moving.
```

Ln 51728, Col 81   100%   Windows (CRLF)   UTF-8

Figure 11: Sample IMU Data Text file

5.4.2 Python

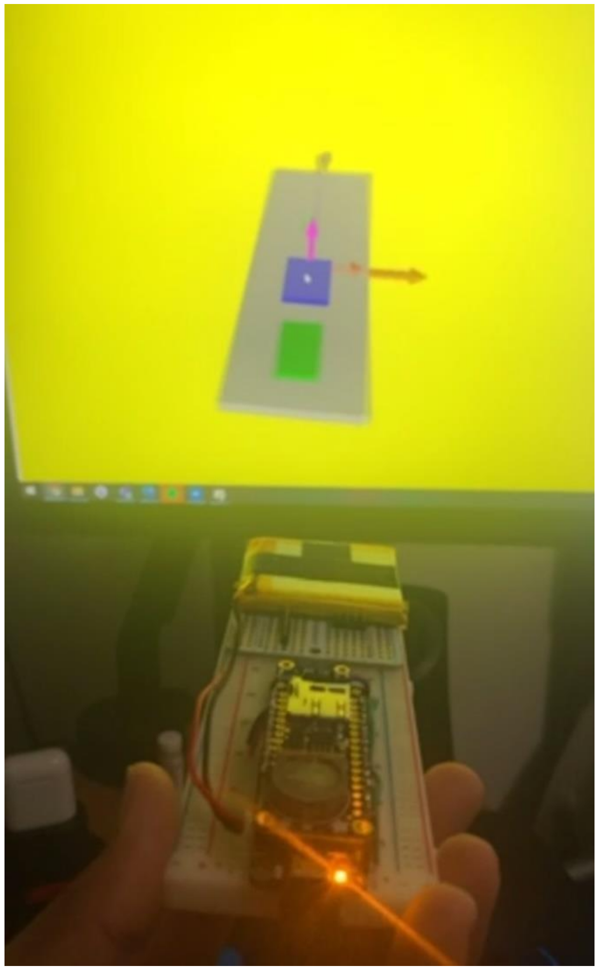


Figure 12: Visual Python Implementation

A Python script is used to convert orientation data from our system into a 3D model to visualize our system (Figure 12). NumPy and Visual Python packages are used to generate the 3D depiction of our system. This method of visualizing the data emphasizes errors or drift from the IMU. Early on in our project, this 3D model was used to tweak the accuracy of the IMU and provide the best estimate of orientation and position.

In addition to the 3D model, simple Python plots (Figure 13) can be used to visual to directly compare the plots to the real-life system. For offline data analysis Python will be used to graph the recorded data from the sporting session to evaluate the performance of an athlete

(Figure 13). For example, the acceleration spikes shown in Figure 13, correspond to the acceleration generated when the user made sharp movements. In addition to looking at graphs, Python will be used to compare the data to accepted performance metrics and thresholds for the sport, if found, abnormalities will be communicated to the athlete in the effort of improving their performance and injury risk. However, like all parts of our system, the end-user can choose a program or method that best suits their needs and comfort. This is easily done as the data is stored on a text file located on the Adalogger's SD Card.

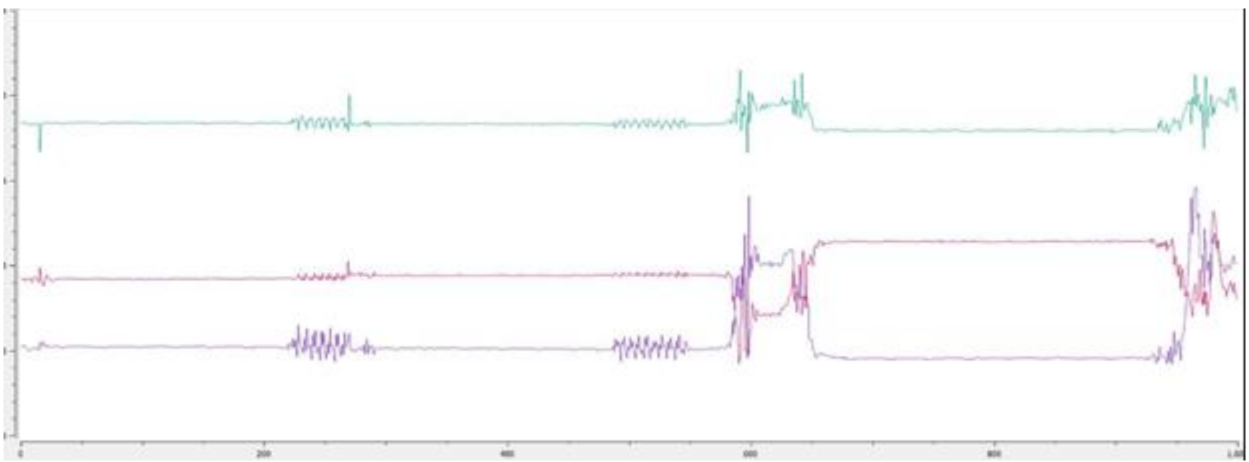


Figure 13: Python Plot of Raw Acceleration Data

5.4.3 Bluetooth Serial Terminal

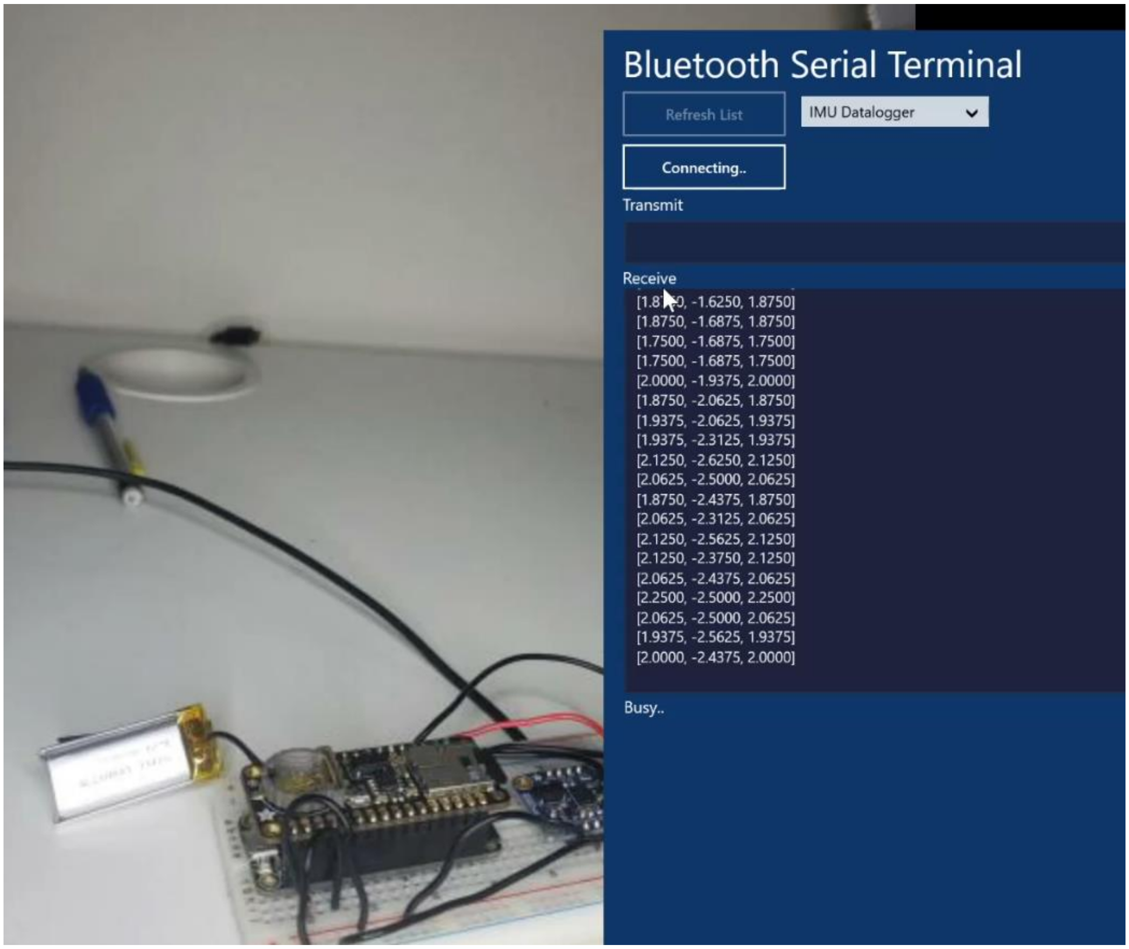
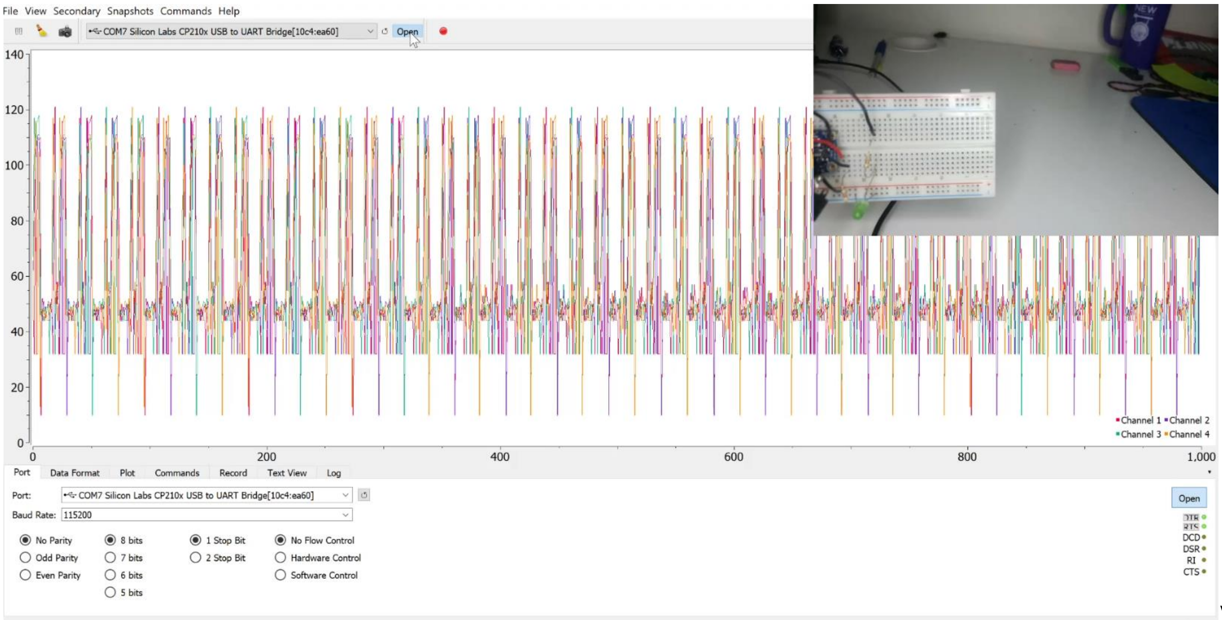


Figure 14: Orientation Data String Received by Wearable IMU Datalogger

If the system is correctly configured to the right operating mode (one push of a button), the Bluetooth Serial Terminal will be able to connect to our system and issue commands. Sending a “1” through the terminal will start the transmission of real-time orientation data to the PC, a “0” will stop the real time transmission of orientation data. If the system is busy datalogging, the Bluetooth Serial Terminal will respond with a “Busy.” As long as the system remains powered, Bluetooth transmission and datalogging can be interchanged constantly. If orientation in data strings is not desired, this data can be viewed in Serial Plotter.

5.4.4 Serial Plotter



\

Figure 15: Plot of Quaternions from Serial Plotter

Here (Figure 15) that same stringed data from the Bluetooth Serial Terminal is plotted in real-time with Serial Plotter. In Figure 15, quaternions are being transmitted in four channels with little endian encoding. Serial Plotter takes the strings of data and plots them over time with 8-bit encoding. Like we mentioned before, the real-time output of data is for surface level analysis. Without serious experience and knowledge with orientation and position data, it is very hard to know what is occurring from this data. The purpose of Serial Plotter and Bluetooth Serial Terminal is to quick way to see your data before you commit to datalogging an entire session. Depending on what type of data you output, ranges, margins, and the channels in Serial Plotter can be adjusted to match the data you wish to output graphically.

5.5 Overall System Integration

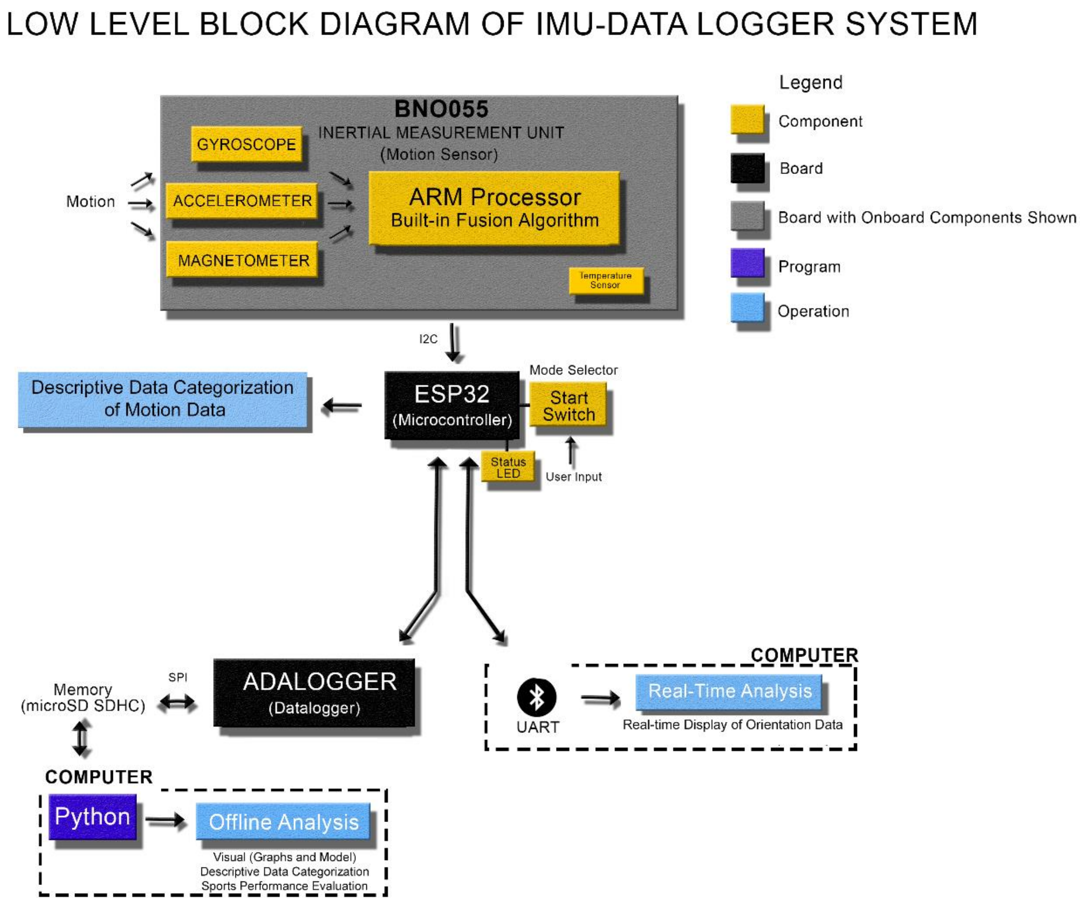


Figure 16: Full System Diagram

The final phase of the project involved integrate the separate functionality stages into one complete system (Figure 16). This entailed compiling the working sketches from the various completed milestones into a single sketch, ensuring all datalogging formats were consistent, and designing the framework to switch between operation modes. When we achieved full system functionality, the software tools running on the main computer could not access the Bluetooth or serial COM port if it were in use. Testing and Validation on the full system integration consisted of a full trial run – both using all the functionality modes in one test and running the system for the entire 5 hours that a durational test could require of the device and inspecting the results. After the five-hour test, 6,256 kB movement data file was generated with the suggested 500mA battery. With a 2,500-mA battery the system ran for fourteen hours uninterrupted with plenty of power left to continue. Overall, this system surpasses our initial goal of a 5-hour use case and can run continuously for several days. However, this is expected as it our system was designed to power additional sensors and the battery was chosen to support that extra power draw.

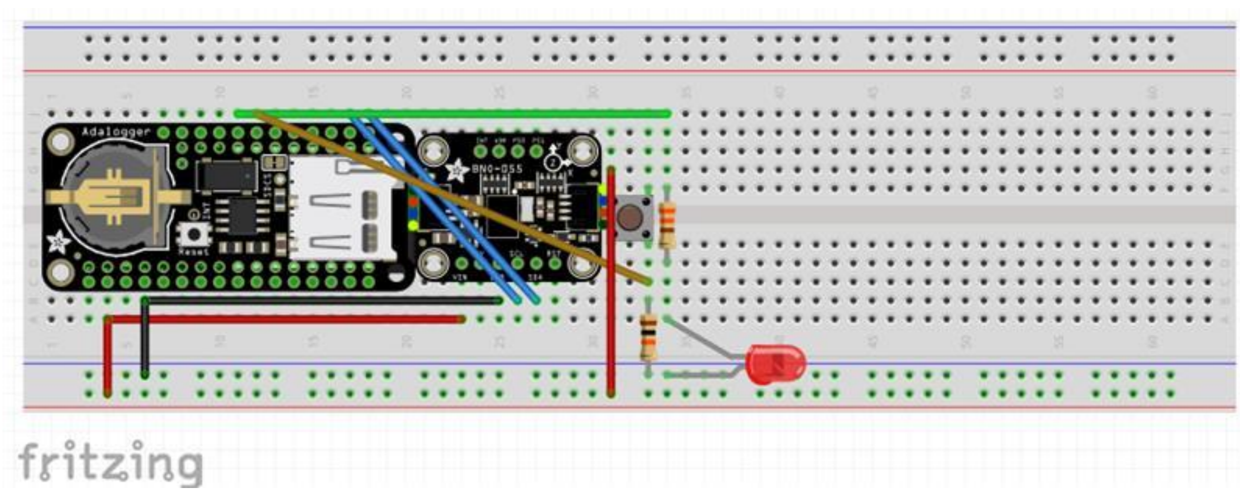


Figure 17: Full System Integration with Button and LED

Overall system integration also includes integration of a simple additional sensor to prove the system's modularity. In the example below (Figure 18), the sensor in question is an ambient light sensor, but potential additional sensors like force or pressure sensors to can be used to compare the athlete's motion data to pressure or force. Additional sensors that record biomedical data, such as a heart rate monitor or pulse oximeter can be added to the system to correlate biological stats to athletic movement. In general, any sensor that can be attached to a microcontroller can be added to this system with few adjustments. Any additional data will be recorded with same I2C SCL line so it can easily be compared to motion. While environmental information such as temperature, ambient light, or barometric pressure may be of interest, users will most likely what data that is directly comparable to the motion recorded. A full demo of system can be viewed at https://youtu.be/I2n_vJK3DQo. This demonstration is of the system in Figure 17, with working datalogging, described movement and data transmission.

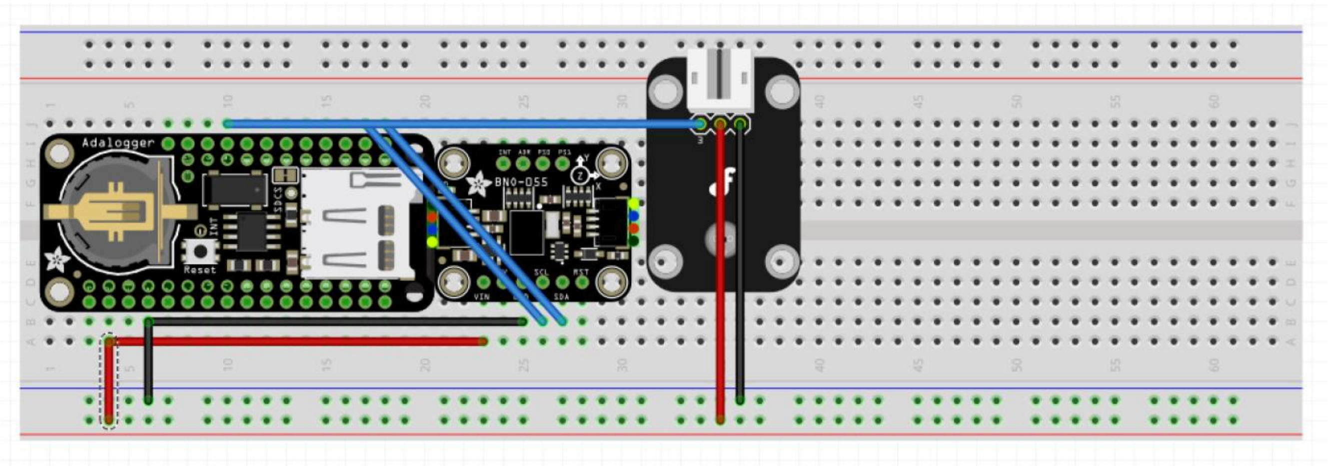


Figure 18: Wiring Diagram for Integration of an Additional Sensor

5.6 Next Steps

The project is functional in the context of testing within team members' own spaces, but there is still potential work to be done. To make the device better suited for the proposed use case; it would be advisable to equip it with some form of protective casing to prevent physical damage. Currently the most complete prototype is mounted on a 51mm x 81mm prototyping board (although this could be cut if it were necessary to conserve further space) and totals 25mm in height. A simple 3D-printed box with a mount would provide physical protection and clearance for the HUZZAH32 headers (9mm in length, but as with the prototyping board could be cut down if needed), would provide protection and a secure position for the sensor to prevent noise in the measurements caused by the device moving within its casing. An example of one such mount, providing the full 9mm of clearance and securing the full size of the soldered prototyping board, is shown in Figure 19 below:

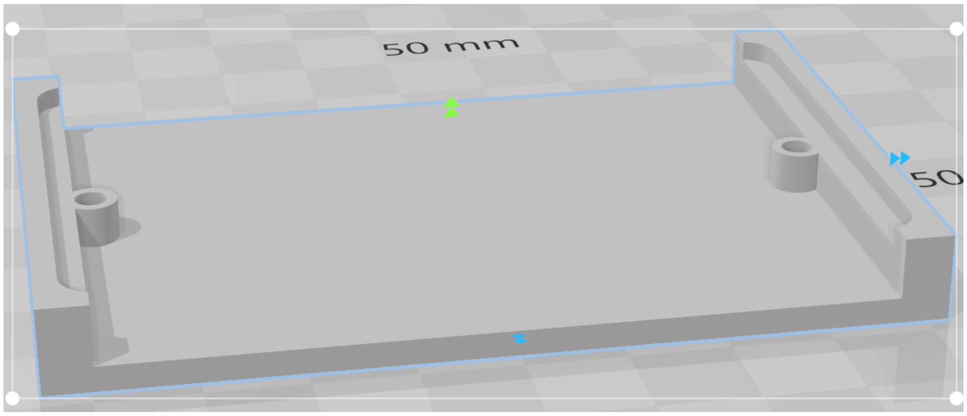


Figure 19: Example of a Mount for the Device Casing [10]

For now, the current state of the project is adequate without a case. Simply soldering the system to a perma-board demonstrates the overall compact and portable design we aimed for. With decent cable management, the button and LED wired, and with a good solder job this is a great base design that can easily handle more sensors (Figure 20).

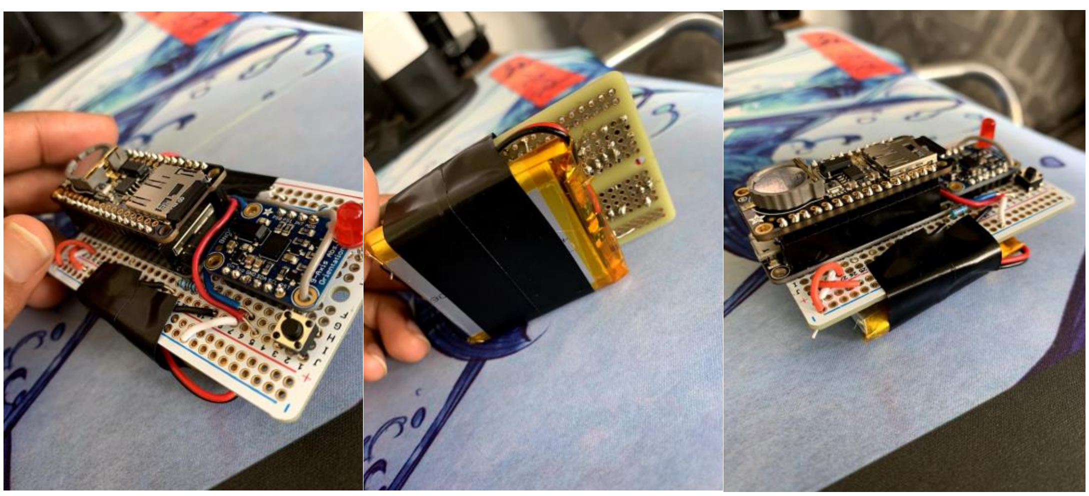


Figure 20: Final Project Soldered to Perma-board

5.7 Critical Review

In comparison to new groundbreaking tech, this system provides no new development regarding the fusing, accuracy, or quality of the movement data of an athlete. This focuses on price, adaptability, and customization. As stated in the motivation, most wearable IMU systems are targeted towards providing an athlete or coach with an “out of the box” solution, rather than something adaptable and useful for other activities and sports. This system is aimed at an end-user who wants more control over their system and without the burden of cost. With a few simple tweaks and additional sensors, this system can be modified to provide adaptive sports data, supporting a variety of disadvantaged groups through customizable systems. The system hopes to provide regular users the benefit of motion analysis that is often afforded to high-level athletes.

6.0 Project Plan

6.1 Technical Milestones

The technical milestones have been further divided since the last progress report to allow for overlapping, modular work towards the various goals. In many cases the former “milestone” has been revised to be a section of milestones to better reflect the individual steps required to accomplish it and to more effectively subdivide the work required for the project.

Table 4: Technical Milestones

Title	Dates	Team Lead	Description
1. Hardware Requirements Identified	November 9, 2020	Ibrahim, Jacob	Requirements for sensor range, interfacing, power consumption, and other features determined and outlined; appropriate hardware selection made.
2. Arduino Outputs Raw IMU Data via Serial	November 30, 2020	Jacob	Program on PC can receive sample IMU data and output useful parameters: position, acceleration, orientation, and visualization.
2(a). Program Performs Signal Conditioning	Optional/ January 31, 2021	Ibrahim	Program on PC can perform signal conditioning that will reduce noise on sample IMU Data. Signal processing on IMU runs

				the risk of altering peaks and timing of events [2] therefore tests will be done to ensure noise reduction does not compromise system integrity.
3. Full System Integration		January 31, 2021 (dependent on hardware shipping)	Ibrahim, Jacob	
	3(a) IMU communicates w/ Microcontroller	January 31, 2021	Jacob	Microcontroller can receive data from IMU and display that data on debug screen.
	3(b) Microcontroller to SD Card (write)	January 31, 2021	Ibrahim	System able to write to SD card. SD card capacity determined to be adequate for 5-hour data recording session based on data rate.
	3(d) IMU to Microcontroller to SD Card	January 31, 2021	Ibrahim	Microcontroller able to write IMU data to SD Card.
4. Wireless Data Transmission successfully completed		February 9, 2021	Harriet/Jacob	System able to start and stop recording wirelessly and provide real-time calibration.
	4(a) Datalogger communicates with PC via Bluetooth.	February 9, 2021	Harriet	Microcontroller can send simple wireless signals to computer.
	4(b) PC Communicates with datalogger via Bluetooth	February 9, 2021	Jacob	Microcontroller can send simple wireless signals to computer.
	4(c) Real-time Calibration data	February 14, 2021	Jacob	IMU and microcontroller able to output real-time orientation data for calibration/testing purposes.
	4(d) SD Card Data Transmission	March 5, 2021	Ibrahim	System able to wirelessly transmit stored data to computer for further analysis.
	4(e) Bidirectional Wireless Communication between Computer and Full System	March 10, 2021	Ibrahim	All wireless signals can be confirmed to be successfully sent to and from the computer and the system.
5. Successful Integration of Hardware and Software		April 11, 2021	All	Full System able to run movement tests, stores, transmits, and outputs position, acceleration,

				orientation. Full-length test (including calibration phase and extended data collection phase) successful.
	5(a) Graphical Processing of IMU Data with PC	April 11, 2021	Ibrahim	Full system able to process and output clear and concise graphs to show relevant data retrieved from IMU.
	5(b) Operational Modes for Datalogging and Transmission	April 11, 2021	Ibrahim	System can switch between calibration and transmission modes successfully.
			Ibrahim	System can switch between calibration and transmission modes successfully.

6.2 Non-Technical Milestones

While all group members contribute to each of the non-technical milestones, one to two team leads have been assigned for each. These team leads are responsible for setting smaller deadlines within the group, delegating unfinished tasks, and leading group meetings pertaining to these milestones.

Title	**Dates** **Start**	**End**	**Team Lead**
Project Proposal Draft	09/19/20	10/18/20	Ibrahim/Jacob
Project Proposal	10/20/20	11/02/20	Ibrahim/Jacob
Progress Report	11/03/20	01/20/21	Ibrahim/Jacob
Final Report Draft	01/21/21	02/26/21	Jacob/Jacob
Video Presentation Draft	01/21/21	02/26/21	Ibrahim/Jacob/Harriet
Final Report	02/27/21	04/14/21	Jacob/Ibrahim
Video Representation	02/27/21	04/14/21	Ibrahim/Jacob

6.3 Gantt Chart

IMU - Data Logger

SYSC 4907

Ibrahim Said & Jacob Nevins
Obehioye Ebozele-Ogbeide

Proj. Start: Sat, 19-Sep-2020
Display Wk: 4

TASK	TEAM MEMBERS	PROGRESS	START	END	DAYS
DELIVERABLES					
Project Proposal Draft	all - Ibrahim	100%	19-Sep-20	12-Oct-20	22
Project Proposal	all - Jacob	100%	10-Oct-20	02-Nov-20	22
Progress Report	all - Ibrahim	100%	15-Oct-20	20-Jan-21	96
Final Report Draft	all - Jacob	100%	30-Dec-20	26-Feb-21	57
Video Presentation Draft	all - Ibrahim	100%	21-Dec-20	26-Feb-21	66
Final Report	all - Jacob	100%	27-Jan-21	14-Apr-21	76
Video Presentation	all - Ibrahim	100%	27-Feb-21	14-Apr-21	45
MILESTONES					
Hardware Identified/Purchased		100%	19-Sep-20	09-Nov-20	50
Arduino Outputs Raw IMU Data via Serial		100%	20-Nov-20	30-Nov-20	9
Signal Conditioning/Fusion Algorithm Tweaks		100%	20-Nov-20	31-Jan-21	71
Full System Integration		100%	20-Nov-20	31-Jan-21	71
IMU communicates w/ Microcontroller		100%	20-Nov-20	31-Jan-21	71
Microcontroller writes to SD Card		100%	20-Nov-20	31-Jan-21	71
IMU writes to SD Card via Microcontroller		100%	20-Nov-20	31-Jan-21	71
Data Transmission Fully Complete		100%	10-Dec-20	09-Feb-21	60
System communicates with PC via Bluetooth		100%	10-Dec-20	09-Feb-21	60
IMU Calibrated via Bluetooth		100%	10-Dec-20	09-Feb-21	60
System Transmits IMU Data to PC via Bluetooth		100%	10-Dec-20	09-Feb-21	60
Software/Hardware Fully Integrated		100%	20-Dec-20	11-Apr-21	111
Graphical Processing of IMU Data		100%	20-Dec-20	11-Apr-21	111
Operational Modes for Datalogging and Transmission		100%	20-Dec-20	11-Apr-21	111
Descriptive Data Categorization		100%	20-Dec-20	11-Apr-21	111

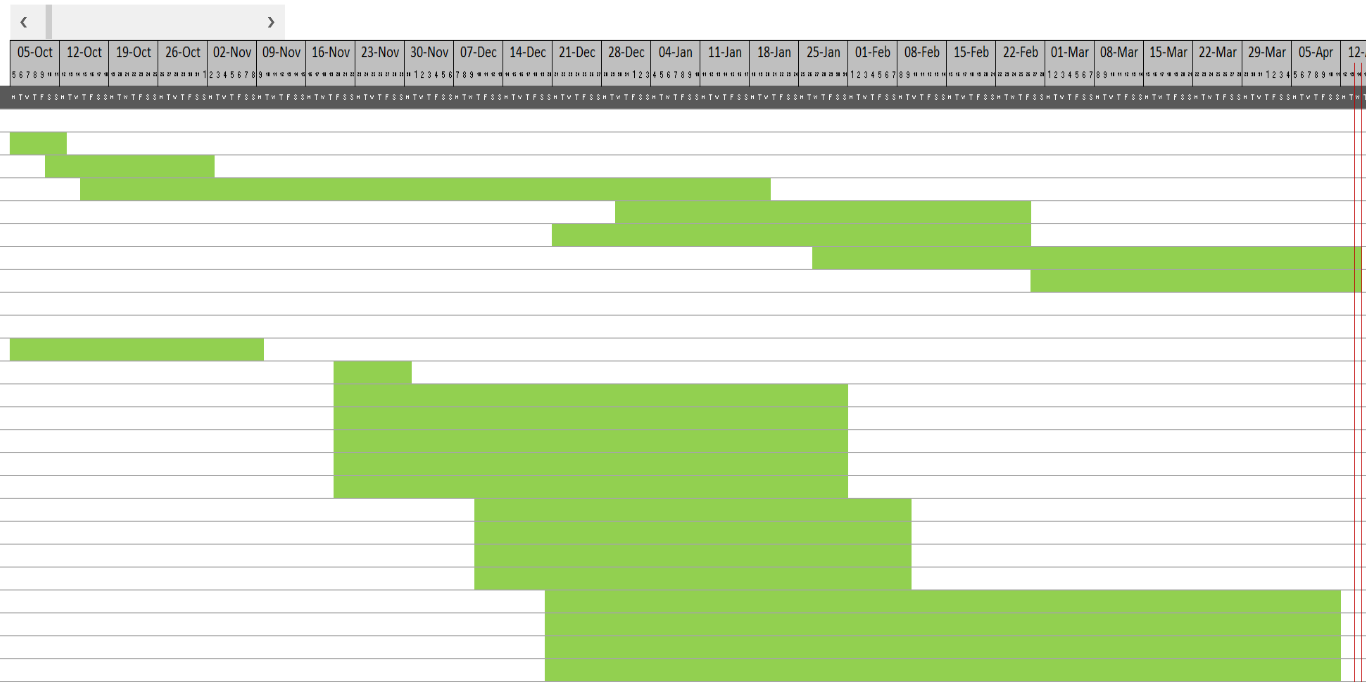


Figure 21: Final Project Gantt Chart

7.0 Project Management

7.1 Meetings

Meetings were done in Microsoft Teams and decided on in advance for the entire semester. Meetings were held in a timely coordinated manner twice each week. One of the two meetings was dedicated to bringing our supervisor up to speed and to get answers to any questions or misunderstandings we encountered during the week. The second meeting was devoted to discussing our overall progress as a team, and to visit and assign technical and non-technical milestones that contribute to the progress of our project. This second meeting was only held between the members of our group. Jacob was assigned to the role of writing down notes based on our weekly meetings with the professor and Obehi and Ibrahim took turns in writing down the weekly agenda in anticipation of upcoming meetings. If additional meetings were required, they would be scheduled into the Microsoft Teams calendar for reference.

7.2 Roles

Each member of the team played a role in ensuring the completion of the project in an efficient and organized manner. Firstly, Jacob and Ibrahim identified the hardware specifications needed to suit the purpose of our project. The hardware requirements were then shown to the professor for verification purposes before the parts were then ordered. When the parts were received, Jacob oversaw verifying that the PC was able to communicate with the IMU via USB to output useful data such as acceleration, position, orientation, and visual representation. After that was completed, Ibrahim oversaw the hardware system integration, which entails the IMU being integrated into the microcontroller then the retrieved IMU data being successfully written and stored in the SD card. When that was verified to be working as expected, Harriet and Jacob worked on the wireless data transmission. Harriet was involved in sending IMU data to a PC via

Bluetooth and Jacob ensured that the PC could issue commands to stop the real-time output of orientation data. The operational modes and conditional threshold tests for described movement changes were done by Ibrahim. Lastly, Ibrahim worked to achieve the successful integration of both the hardware and software parts involved in our system.

8.0 Relevance to Program

The system's intended purpose in the field of sports monitoring is generally relevant to the field of Biomedical Engineering (the field of study of all team members). It involves working with sensors, signal processing, and programming, subjects addressed in numerous other courses in the degree program. It also addresses many of the CEAB Graduate attributes: the project furthers a knowledge base in the aforementioned areas highly relevant to biomedical engineering, requires Problem Analysis, Investigation, and Design to examine the use case and design towards it, makes use of a variety of Engineering Tools, and like any group project of this nature, requires Individual and Teamwork, Communication Skills, Professionalism, and Project Management.

9.0 Health and Safety

The project was conducted in an extremely safe manner. Extra caution was taken during the soldering process of the components of our system, which includes the BNO055 IMU, ESP32 Adafruit microcontroller, and the corresponding Adafruit feather, which was attached on top of the microcontroller, to ensure no burn-related injury was sustained and the areas which the soldering iron met were left unaffected. Also, an open window and fans were used to push out fumes from soldering. Due to the COVID-19 regulations, we each carried out all assignments regarding building our system individually in our various houses to adhere to social distancing rules. In situations where we ran into some difficulties and we had certain questions and concerns regarding our project, we communicated with one another via Microsoft teams. In this way, we were able to accomplish the completion of our project while staying safe and respecting the COVID-19 rules and regulations at the same time.

References

[1] V. Camomilla, E. Bergamini, S. Fantozzi, and G. Vannozzi, “Trends Supporting the In-Field Use of Wearable Inertial Sensors for Sport Performance Evaluation: A Systematic Review,” *Sensors*, vol. 18, no. 3, p. 873, 2018.

[2] Y. Adesida, E. Papi, and A. H. Mcgregor, “Exploring the Role of Wearable Technology in Sport Kinematics and Kinetics: A Systematic Review,” *Sensors*, vol. 19, no. 7, p. 1597, 2019.

[3] S. L. Colyer, M. Evans, D. P. Cosker, and A. I. T. Salo, “A Review of the Evolution of Vision-Based Motion Analysis and the Integration of Advanced Computer Vision Methods Towards Developing a Markerless System,” *Sports Medicine - Open*, vol. 4, no. 1, 2018.

[4] A. M. Gal, T. Douglas, A. D. Chan, and D. C. Hay, “Potential for autonomous linear, curvilinear, and phase detection in natural context para skating using IMU sledge motion data,” 2020 IEEE International Symposium on Medical Measurements and Applications (MeMeA), 2020.

[5] D. A. James, “The Application of Inertial Sensors in Elite Sports Monitoring,” *The Engineering of Sport 6*, pp. 289–294, 2006.

[6] International Paralympic Comittee, “Ice Sledge Hockey Rules 2014-2018.” [Online]. Available: https://www.paralympic.org/sites/default/files/document/140728153826737_2014_07_28_IPC_ISH_Rulebook_2014-2018.pdf. [Accessed Jan. 18, 2021]

[7] “PS3439 Blue Trident factsheet.” [Online]. Available: https://imeasureu.com/wp-content/uploads/2019/06/PS3439_Blue-Trident-factsheet_Final-1.pdf. [Accessed: 20-Feb-2021].

[8] “MMC,” *MbientLab*, 16-Feb-2021. [Online]. Available: https://mbientlab.com/metamotionc/. [Accessed: 19-Feb-2021].

[9] *Help Centre*. [Online]. Available: https://support.tritonwear.com/. [Accessed: 25-Feb-2021].

[10] Modified from a design by thingiverse.com user DaveGun https://www.thingiverse.com/davegun/designs