ECE 445: Senior Design Laboratory

Running Cadence Monitor Belt

Team 5

Alexander Jin, Dante Vasudevan, Nick Bergerhouse

Professor: Viktor Gruev

TA: Koushik Udayachandran

Spring 2024

Abstract

The running cadence monitor belt was designed to provide a runner with hands-free, eyes-free tracking of their cadence. While other solutions such as smartphones and smartwatches can track cadence, our device seeks to limit the distractions placed on the runner by using a haptic feedback system for hands-free notification. This combined with other systems such as a step detection algorithm using data from an integrated Inertial Measurement Unit (IMU) allows seamless pace correction for runners when their cadence falls outside of their desired range.

Contents

1 Introduction

1.1 Problem and Solution

Running cadence is the number of steps a runner takes per minute while running, commonly measured in strides per minute (SPM). It is a useful measurement for runners as it can provide insight into efficiency, form, and stride length. An ideal cadence for most runners typically falls into the range of 170 to 180 SPM although this is dependent on height and pace.

Currently there are already products on the market that can measure running cadence. For example, most “running watches” have cadence as an included measurement. However, it can be cumbersome for runners to constantly switch through display screens to monitor multiple data points at the same time such as pace, heart rate, distance, and cadence. Furthermore, unless a runner is running with their arm locked in front of them, continuous monitoring of cadence is impossible with a running watch. Other products take a different approach such as the foot-mounted ARION Footpod non-GPS 1.0[1] and Stryd[2]. These products can track the cadence throughout the run in much the same way that a running watch would, but they can not provide that information to the runner without the use of a watch or smartphone. In both the watch and foot-mounted solution, there is a lack of a product that provides easy, hands-free haptic feedback to the runner informing them when their cadence falls outside of the ideal cadence range.

The product consists of a lightweight, belt-mounted device consisting of an internal measurement unit (IMU) to gather movement data for step detection, Microcontroller Unit (MCU) for step detection calculation, and haptic feedback systems to interface with the user. A running mean time between a certain number of previous steps is used to calculate the runner’s current cadence. Based on the measured cadence, the microcontroller controls vibration motors to create haptic feedback, which informs the user based on vibration patterns in real time how to adjust their cadence to achieve perfect running efficiency. The device itself is mounted on the user’s waist, as this is already a popular spot for runners to store items, such as phone mounts or fanny packs. This also increases user comfort by keeping the device clear of the front, where there may be hand movement. The system is solely be powered by a Lithium-Polymer (LiPo) battery [3] situated on the belt. Users are able to adjust their target cadence from a default of 180 to a floor of 120 in increments of 5.

The microcontroller subsystem consists of an ESP32-S3 microcontroller along with supporting circuitry, which handles step detection, cadence calculation and haptic feedback activation decisions. Due to difficulties of extracting consistent output from the microcontroller during development, software was moved to an ESP32-S3 development board.

The IMU subsystem consists of a BNO086 Inertial Measurement Unit [4] and supporting circuitry. The IMU sends linear acceleration data to the microcontroller via an exclusive SPI bus interface. Due to difficulties communicating with the selected IMU during development, algorithm verification was performed using a BNO055 IMU development board.

The Haptic Feedback subsystem contains vibration motors activated via a control signal from the microcontroller to indicate a suboptimal cadence to the user. A pushbutton is present to allow cadence adjustment. During development, a planned second pushbutton to allow haptic feedback strength control was removed.

All three boards are mounted onto a flexible waist belt. The battery pack is located underneath the Battery Management System (BMS), near the IMU board. The vibration motors are attached to the Haptic Feedback Board shell to both generate noise and propogate vibration to the user. The entire system, while mounted on the belt, are enclosed in individual 3D printed plastic shells.

1.2 Block, Physical Diagram

Figure 1.1 shows the Block Diagram of the belt. Figure 1.2 shows the Physical Diagram of the belt. Of particular note is the location of the BNO086 IMU sensing unit on the side of the left leg, in order to locate it near the user's hip. The spacing between the left/right boards and the front of the belt , represented by buckle halves, has been shortened in the interest of ease of display.

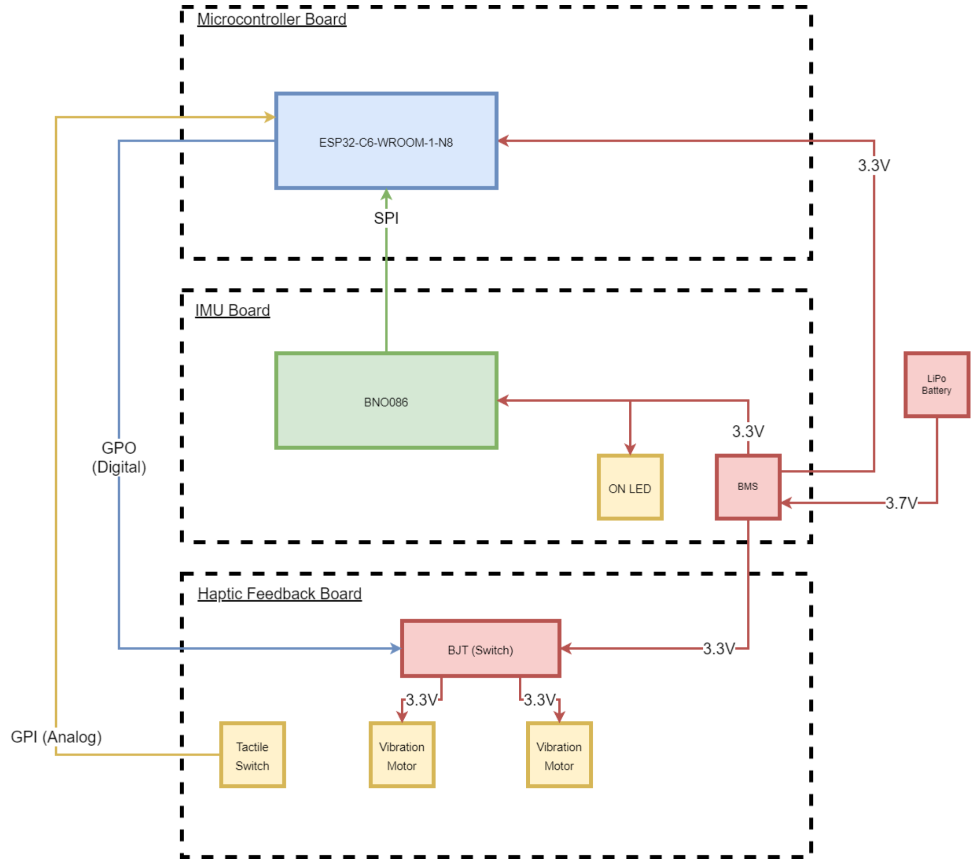

Figure 1.1: Block Diagram

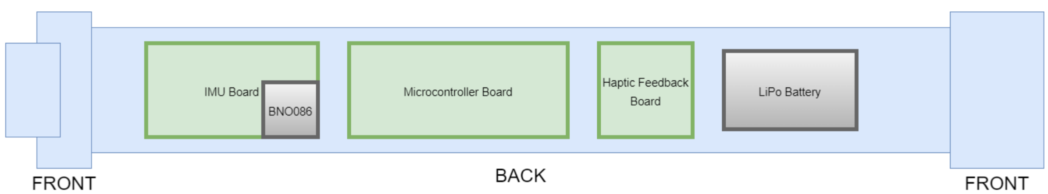

Figure 1.2: Physical Diagram

1.3 High Level Requirements List

1. The product shall use an IMU to derive an SPM cadence metric. The default SPM goal is 180.

2. The product shall notify the user of a measured SPM outside of the set limit (+/- 10) of a user-adjustable goal SPM via vibration motors located on the device.

3. The product shall have the goal SPM be user adjustable in increments of 5 through a tactile switch located on the Haptic Feedback Board.

4. The product shall be solely powered by a portable 3.7V LiPo battery capable of 1.5A current output.

2 Design

2.1 Microcontroller Board

2.1.1 Design Procedure

The Microcontroller Board houses the microcontroller, which is responsible for taking in metrics from the IMU and calculating a cadence as well as making decisions based off of that cadence. An ESP32-S3 microcontroller was chosen for its power, compatibility with Arduino, and potential for a bluetooth app stretch goal due to its integrated antenna. Figure 2.1 shows a general block diagram of the subsystem.

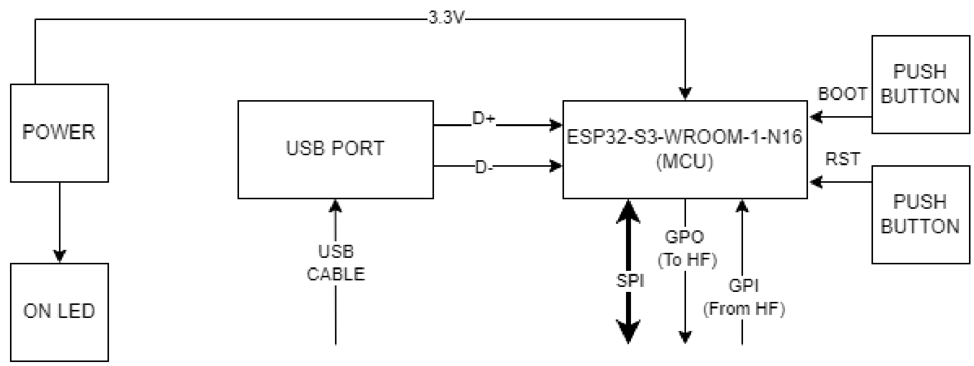

Figure 2.1: Microcontroller Board Block Diagram

2.1.2 Design Details

This board contains the ESP32 microcontroller. It is connected to the IMU board through an exclusive Serial Peripheral Interface (SPI) data connection, as well as to the Haptic Feedback Board through several GPIO pins to receive button inputs and activate the vibration motors. The SPI interface it uses to connect to the BNO086 IMU is located on the "HSPI" interface pins. Figure 2.2 shows a detailed schematic of the subsystem. Of note in the schematic is the present of pinouts for the ESP32-S3 due to the debugging procedures that were carried out on this subsystem.

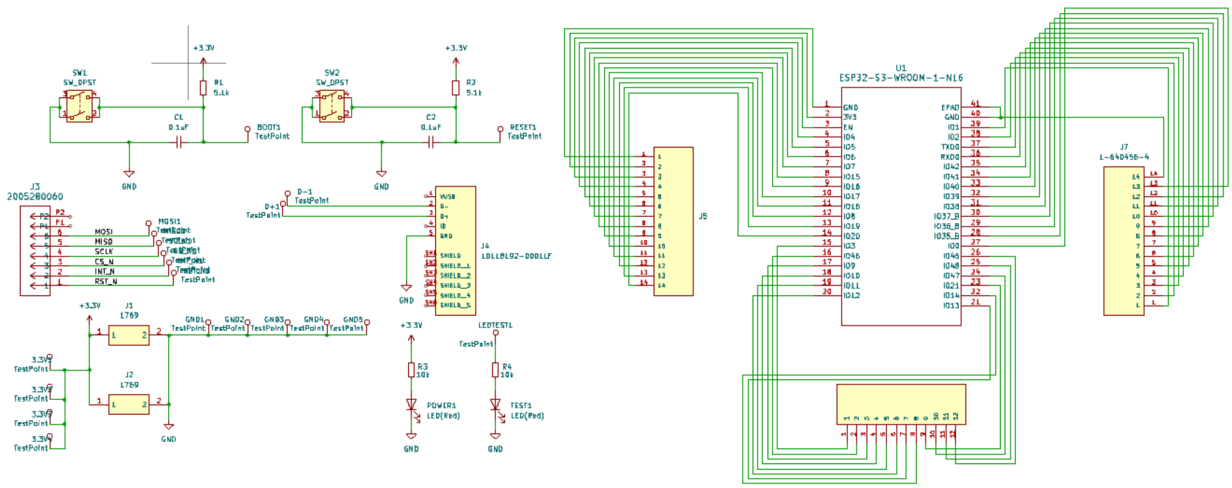

Figure 2.2: Microcontroller Board Circuit Schematic

The microcontroller receives data from the IMU and calculates the SPM of the user. Based on the user's SPM compared to a target cadence, it alerts the user to fix their SPM by activating different vibration patterns on the vibration motors, which are further described in the Haptic Feedback Board section. It is also capable of receiving tactile switch inputs from the Haptic Feedback Board to alter the goal SPM in increments of +/-5. The Microcontroller subsystem features an ON LED, as well as a BOOT/RESET circuit to enable programming of the microcontroller. This feature is only needed for development purposes.

2.2 Algorithm

2.2.1 Design Procedure

The algorithm is responsible for converting the raw data from the IMU into a usable SPM value, comparing it to the target cadence, and sending the correct digital signals to the haptic motors. It compares this SPM value to the user adjustable goal SPM. A sample IMU linear acceleration dataset collected from a smartphone IMU mounted to the hip of a runner is used to illustrate design choices.

2.2.2 Design Details

The cadence algorithm is the most complex portion, as design choices were needed to handle step detection, noise filters, and cadence calculation. The step detection algorithm relies on observing peaks in the linear acceleration magnitude. With each step, the IMU will experience momentary acceleration impulses, which will peak the net acceleration noticeably above the mean. Given the fixed nature of the IMU relative to the user's leg and hip, we elected to discount both the Z and Y dimensions (side to side and up and down, respectively). The Z dimension is not strongly related of the movement of a leg during running, and the Y dimension was much less indicative of steps vs. the X dimension (forwards and backwards). This can be seen in Figure 2.3. If the aforementioned peaks have a larger magnitude than a step threshold, located at one standard deviation from the mean, a step is counted.

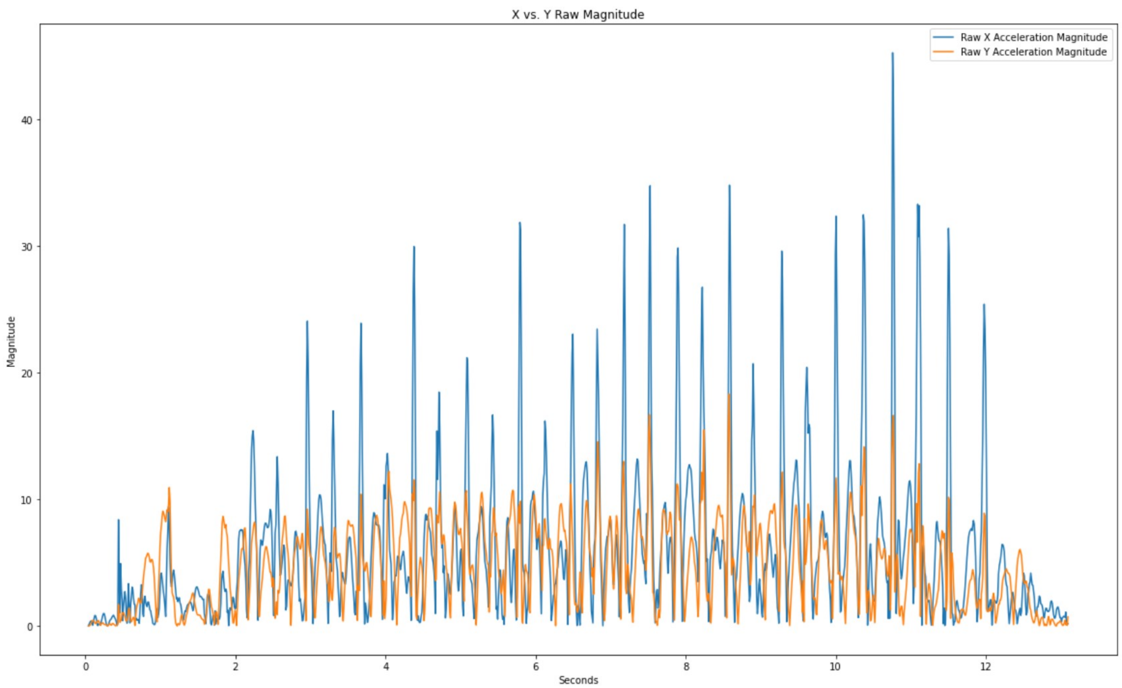

Figure 2.3: X vs Y Unfiltered Linear Acceleration Magnitude

After collecting raw IMU data, we built a second-order lowpass butterworth filter to handle noise filtration. Equation (2.1) shows the continuous transfer function for an nth-order butterworth filter:

$$H(s) = \frac{1}{\sum_{k=1}^{n} \frac{a_k}{\omega_0^k} s^k} \qquad \text{where} \qquad a_{k+1} = \frac{cos(k\gamma)}{sin((k+1)\gamma)} a_k \tag{2.1}$$

We needed a filter with a sharp cutoff, as our useful data had a similar frequency as our noise. Higher order filters are better for this purpose. However, we wanted a small delay, to provide an accurate live cadence to the user. Lower order filters are needed to achieve this. We found that a second-order filter is the perfect balance between the two conditions. Figure 2.4 shows the comparison between the filtered and unfiltered net linear acceleration magnitude.

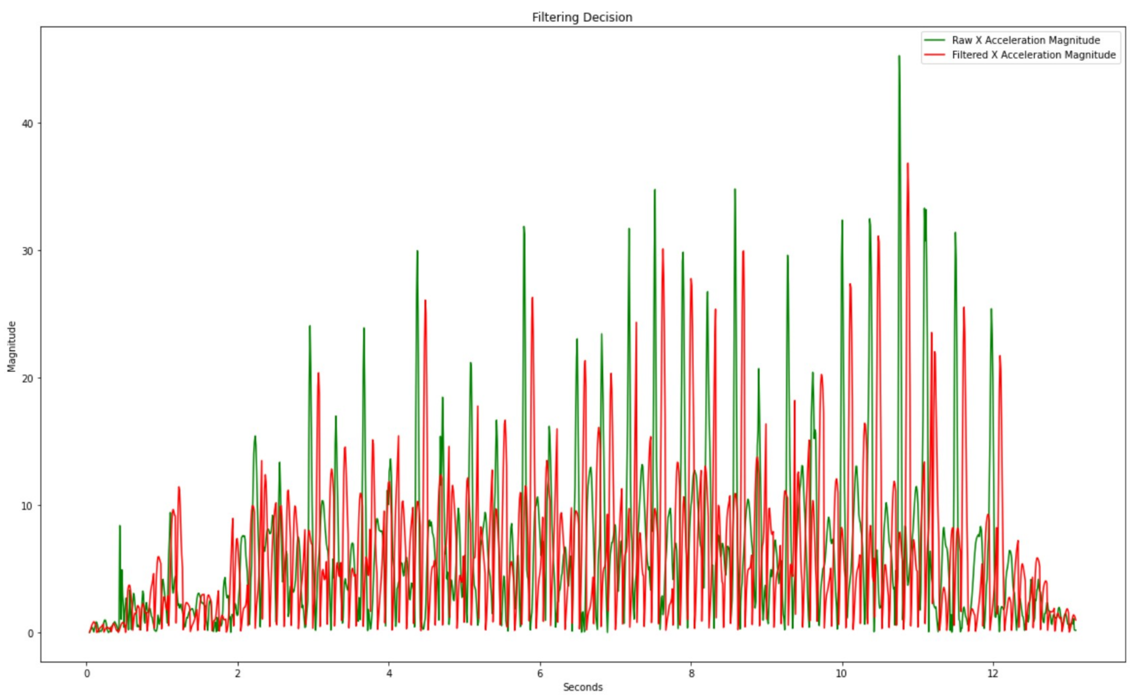

Figure 2.4: Filtered vs Unfiltered Linear Acceleration Magnitude

We found that the noise was not significant enough to impact our steps. Due to the similar frequency between the noise and our useful data, the filter only attenuated our data, resulting in less accurate step detection and a useless delay. For those reasons, we chose to remove the filter from our algorithm.

To determine cadence, we created a running SPM value by dividing the current number of steps by the time. The more steps that were taken, the more accurate the cadence became. Figure 2.5 shows the running cadence value algorithm being fed with steps at a rate of 180 SPM compared with a target cadence of 180 SPM. As we can see, our cadence value arrives within 5 SPM of the target within 3 seconds, and can become accurate at a faster rate at higher SPM values.

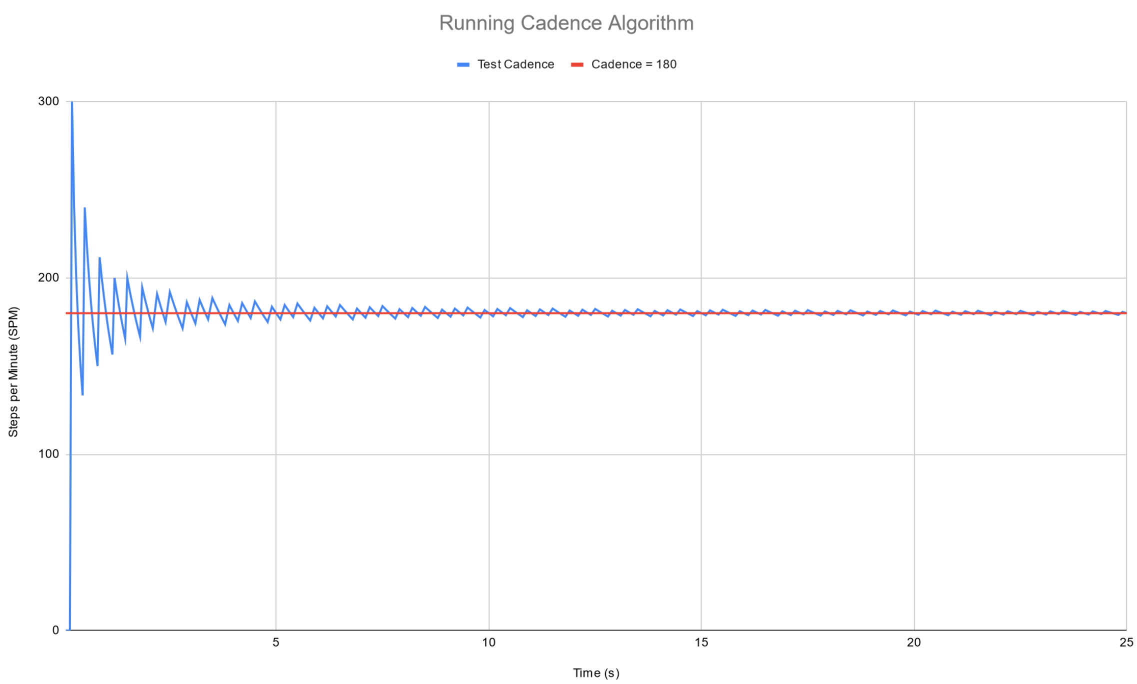

Figure 2.5: Cadence Inaccuracy Settlement

Another design consideration is the automatic pause when the user is not actively running. There are many scenarios in a run in which the user might stop, for example waiting at a stop-light. To handle this, we implemented a condition that if the user's cadence was smaller than 120 SPM (walking pace), the haptic feedback would not activate. This however posed a problem to our running cadence value, as the value could get skewed by an extended period of low values. Hence, we reset the cadence calculation every 10 seconds.

We designed the haptic feedback to provide different patterns based on the state of the user's SPM compared with the target SPM. If the user's SPM was larger than the target range, the device would activate for one long burst. If the user's SPM was smaller than the target range, the device would activate for two short bursts.

User customization is a very important design consideration, as not every user might want the default target cadence programmed. We programmed a tactile switch, that if briefly pressed, the target cadence would reduce by 5 SPM. However, if the switch was pressed for longer than 2 seconds, the target cadence would increase by 5 SPM.

2.3 IMU Board

2.3.1 Design Procedure

This board contains the BNO086 IMU. The Linear Acceleration Vector detected by this IMU is routed to the ESP32 by a SPI connection. Figure 2.6 shows a general block diagram of the subsystem.

The BMS is also adjacent to this board, specifically a ZIO LIPO BATTERY MANAGER[5] BMS. The BMS contains the system power switch, as well facilitating recharging of the LiPo battery. 3.3V power from the BMS is routed to the rest of the system through a connection to the IMU board. The IMU subsystem also includes an ON LED.

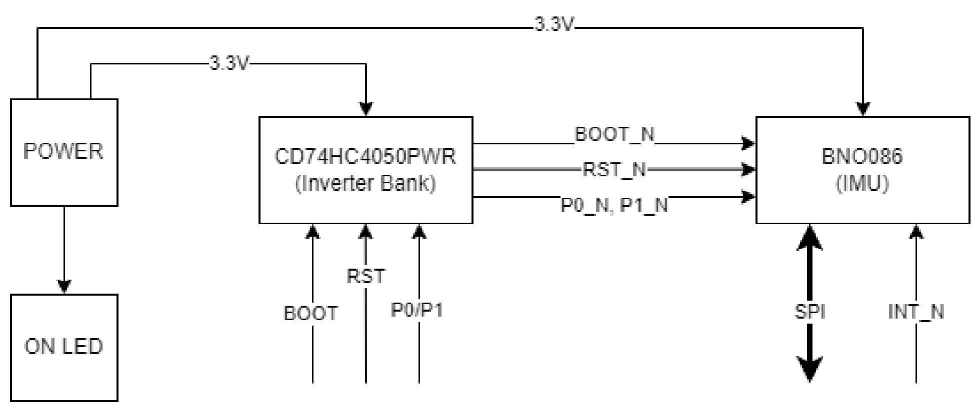

Figure 2.6: IMU Board Block Diagram

2.3.2 Design Details

The IMU was configured to communicate over SPI via configuration of the PS1/PS0 pins. It was additionally configured to run off of an external 32.768kHz crystal oscillator instead of its internal oscillator, to ensure sample accuracy. The majority of the PCB is devoted to pull-up/pull-downs on various BNO086 control signals, such as disabling the firmware flashing feature (BOOT_N). Figure 2.7 shows a detailed schematic of the subsystem. The crystal oscillator is marked as X2 on the schematic.

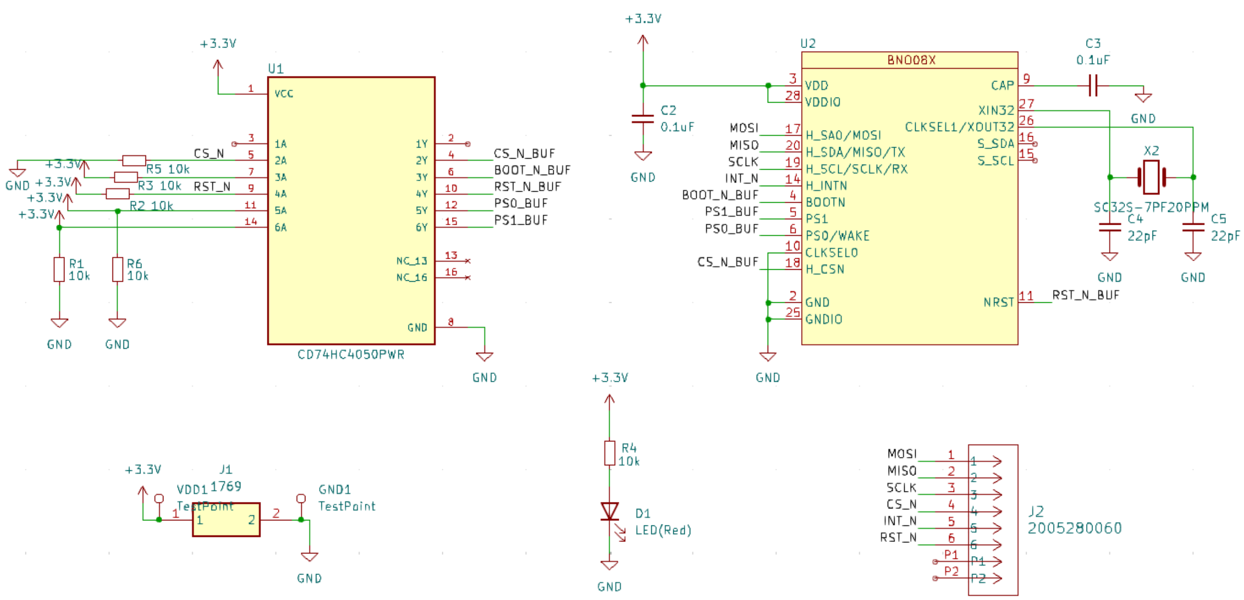

Figure 2.7: IMU Board Circuit Schematic

The SPI bus itself facilitates connection to the Microcontroller Board through the usage of a ribbon cable connector (J2). The IMU Board serves as the main power distributor from the BMS to the rest of the subsystems, through a JST-2 connector (J1).

2.4 Haptic Feedback Board

2.4.1 Design Procedure

The Haptic Feedback Board can be divided into essentially two switches. One switch closes when receiving a GPO from the MCU and the other is a tactile button switch that sends a GPI to the MCU. The GPO switch was relatively trivial; a tactile button connected to 3.3 V with an appropriate pull-down resistor and debouncing capacitor. This switch allows the user to adjust the goal SPM; a single press will lower the goal SPM by 5 from a maximum of 180, and a long press will raise the goal SPM by 5. Figure 2.8 shows a general block diagram of the subsystem.

The main objective of the Haptic Feedback Board is to house vibration motors to enable hands-free communication with the user. These motors are not be able to activate independent of one another, rather different vibration patterns indicate to the user whether they should decrease or increase their SPM. A single long pulse denotes an SPM higher than the goal. Two short pulses denote an SPM lower than the goal. Figure 2.9 shows a detailed schematic of the subsystem.

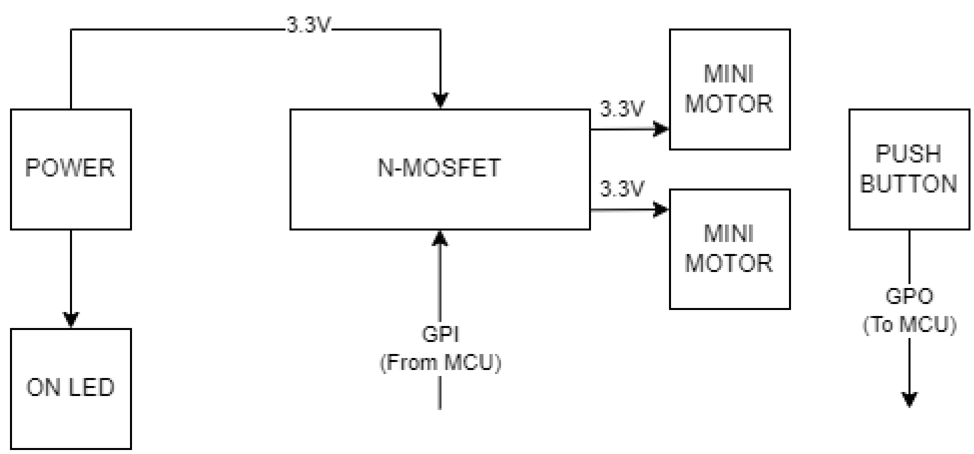

Figure 2.8: Haptic Feedback Board Block Diagram

2.4.2 Design Details

The less trivial part of this board was the coin-type vibrational motors' power switch. These motors were chosen for their relative strength-to-size ratio, as well as for being self-contained coin-type motors. The motors we chose were very small and had low power consumption so immediately we opted for a single transistor solution. The decision then became whether a BJT or MOSFET would be more appropriate. We decided that since our switching input would be a digital high from the MCU, an N-MOSFET [6] would be more appropriate as it is a voltage-controlled device compared to the current-controlled BJT. Its gate is connected to the GPO signal from the Microcontroller Board, and its drain connected to the power for two SEEED STUDIO 2.0MM MINI[7] vibration motors.

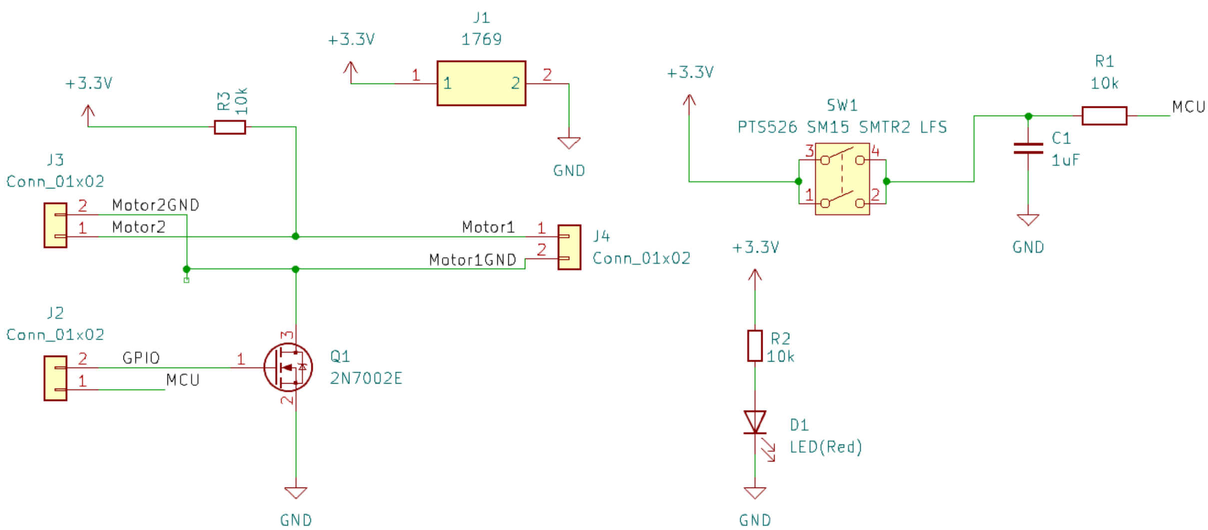

Figure 2.9: Haptic Feedback Board Circuit Schematic

Of particular note in Figure 2.9, J1 is for the JST power connector, J2 are through-holes for IO wires to the MCU, and J3 and J4 are through-holes for the vibrational motors.

3 Verification

3.1 Microcontroller Board

The full Requirements and Verification (R&V) table can be found in Appendix A.

There were three main requirements the Microcontroller Board needed to fulfill. The microcontroller needed to be powered entirely by the Battery and BMS. It needed to have the SPI communication with the IMU Board. And finally, it needed to be able to send digital signals to the Haptic Feedback board.

The first requirement was easily verified with a digital multimeter. By simply probing between the ESP32 3.3V and Ground lies, we showed the MCU board was completely powered with it. The third requirement was demonstrated by supplying a HIGH output to the GPIO_4 pin, which was connected to the haptic motor. When that was done, the haptic motor successfully activated. After later switching to the development board, this continued to be the case. When the user's cadence was below the threshold range, the haptic motors successfully activated in two short bursts. When the user's cadence was above the threshold range, the haptic motors successfully activated in one long burst.

The second requirement was unsuccessful. We were unable to set up communication between the BNO086 IMU, and our microcontroller board as the microcontroller programmed but refused to run programs consistently. Despite extensive debugging including isolation of the microcontroller with only a boot circuit and power, a fix could not be established. A switch to the development board, and a swap to the BNO055 IMU was necessary to establish communication and recieve the necessary information for the algorithm.

3.2 Algorithm

The full R&V table can be found in Appendix B.

There were three main requirements the Algorithm needed to fulfill. The algorithm needed to accurately determine the User's Cadence in units of Steps per Minute (SPM). It also needed to initiate the correct digital signal sequence for the Haptic Feedback Board, based on the user's cadence. The final requirement of the algorithm is to adjust the target cadence based on the manner in which the tactile switch is activated.

The first requirement was verified by collecting IMU data from a 15 second run, and feeding it into the algorithm. Figure 3.1 shows the algorithm being applied to a sample data set. This is the same sample data set as mentioned during the Design section for the algorithm. We can see that the algorithm indeed counting steps every time the acceleration peaks above the threshold. The accuracy of the cadence value was later corroborated through the use of a metronome. When shaking the IMU according to the metronome's Beats per Minute value, the cadence returned with close to if not the same value in Steps per Minute.

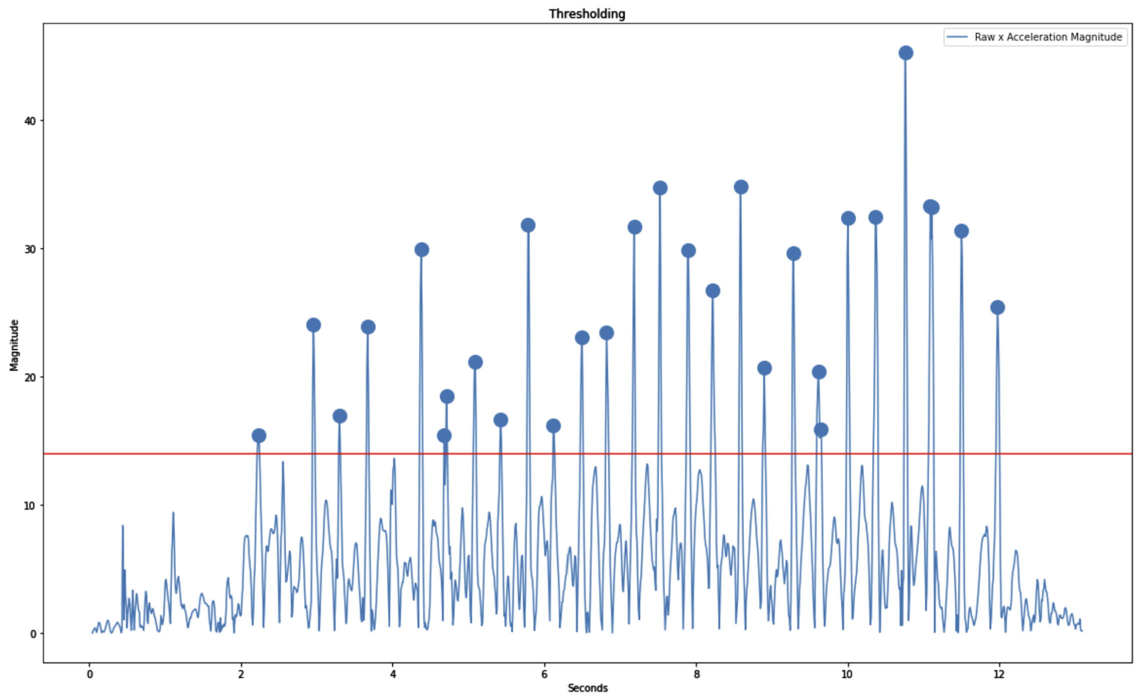

Figure 3.1: Step Detection Thresholding

The second requirement can be broken down into three subrequirements, all of which were verified. The first subrequirement was to initiate a long activation signal for the haptic motors whenever the user's cadence was above 190SPM, the upper bound of the target range. When shaking the IMU at a very fast rate, substancially faster than 190SPM, the haptic motors activated in a long burst, thus confirming the algorithm worked. The next subrequirement was to initiate two short activation signals for the haptic motors whenever the user's cadence was below 170SPM, the lower bound of the target range, but above 120SPM, which is walking speed. Shaking the IMU according to a metronome set at 140BPM activated the haptic motors in two short bursts, thus verifying that portion of the requirement. The final subrequirement was to initiate no activation signal below 120SPM. Every 10 seconds is when the algorithm would provide a digital signal to the motors, so after leaving the IMU untouched for longer than 10 seconds and the haptic motors did not activate, the final subrequirement was verified.

The third requirement was to increase or decrease the target cadence by 5SPM whenever the tactile switch was activated, depending on if it was activated for longer or shorter than 2 seconds, respectively. This was verified by observing the serial monitor when the tactile switch was activated, and seeing the printed target cadence value change in real time.

3.3 IMU Board

The full R&V table can be found in Appendix C.

There were two main requirements the IMU Board needed to fulfill. The IMU needed to be powered entirely by the Battery and BMS. It also needed to supply the necessary metrics used for the cadence algorithm to the microcontroller unit, including linear acceleration vectors and orientation and angular velocity vectors.

The first requirement was easily verified with a digital multimeter. By simply probing between the BNO086 3.3V and Ground lies, we showed the IMU board was completely powered with it. However, the second requirement was unsuccessful. We were unable to set up communication between the BNO086 IMU and both the microcontroller board, and later the development board, and use it to supply the necessary metrics needed for the algorithm.

It was only after switching to a different IMU, a BNO055 development board[8], were we able to set up communication and successfully supply the necessary metrics to verify the Algorithm. This was necessary as communication between the IMU and Microcontroller was very unreliable at best, as stated before.

3.4 Haptic Feedback Board

The full R&V table can be found in Appendix D.

There were three main requirements the Haptic Feedback Board needed to fulfill. The haptic feedback board needed to be powered entirely by the Battery and BMS. It needed to supply the coin vibration motors with 3.3V from the power line whenever microcontroller digital output signal was High. And finally, the tactile switch needed to provide a digital signal to the GPI pins of the microcontroller

The first requirement was easily verified with a digital multimeter. By simply probing between the board's 3.3V and Ground lies, we showed the MCU board was completely powered with it. The second requirement was demonstrated through the activation of the vibration motors whenever the signal fed from the microcontroller was high. After later switching to the development board, this continued to be the case. When the user's cadence was below the threshold range, the haptic motors successfully activated in two short bursts. When the user's cadence was above the threshold range, the haptic motors successfully activated in one long burst.

The third requirement was also successful. Using a digital multimeter, we found that the GPI_5 pin on the Microcontroller Board went to 3.3V whenever the tactile switch was pressed. This was later corroborated after switching to the development board, where we were able to demonstrate the changing of the target cadence any time the tactile switch was pressed.

4 Cost

4.1 Cost Analysis

Equation (4.1) shows the estimated cost of labor for the product during development had group members been paid hourly wages. This cost is purely theoretical. In calculating labor costs, the average starting hourly rate of a UIUC BSc Electrical Engineering graduate is roughly $42 per hour equivalent, based on a salary of $87,000 per year. Accurate logs of hours worked were not made, however a rough assumption of an average of 10 hours per week, per person for 10 weeks is used.

$$Labor = \frac{\$42}{hour} * 10\ hours * 10\ weeks * 2.5 * 3\ people = \$31,500 \tag{4.1}$$

Table 4.1 shows the cost of components used to construct the product, in order to represent a theoretical per-unit cost. This cost does not take into account bulk part order discounts. All equipment used such as Oscilloscopes, Digital Multimeters (DMM), Soldering Irons, Soldering Supplies, Hot Air Stations, etc. are considered "Free" in cost due to being supplied by the Senior Design Lab.

Part	**Function**	**Online Cost**	**ECE Cost**	**Qty**	**Cost**
Zio LiPo Battery Manager	BMS	$19.90	$19.90	1	$19.90
HXJNLDC	LiPo Battery	$14.99	$14.99	1	$14.99
Mini vibration motor 2.0mm	Vibration Motor	$1.20	$1.20	2	$2.40
BNO086	IMU	$15.97	$15.97	1	$15.97
ESP32-C6-WROOM-1-N8	MCU	$3.50	$0.00	1	$0.00
2N7002ET7G	MOSFET	$0.21	$0.21	1	$0.21
PTS526 SM15 SMTR2 LFS	Tactile Switch	$0.14	$0.14	3	$0.42
LTST-C190TBKT.PDF	SMD LED	$0.34	$0.00	3	$0.00
CD74HC4050PWR	Buffer	$0.58	$0.58	1	$0.58
0603N220J500CT	Capacitor	$0.10	$0.00	2	$0.00
CL21F104ZAANNNC	Capacitor	$0.10	$0.00	2	$0.00
SC32S-7PF20PPM	Oscillator	$0.38	$0.38	1	$0.38
485-4714	JST-2 Jumper	$0.95	$0.95	2	$1.90
485-1769	JST-2 Header	$0.75	$0.75	4	$3.00
WM24188CT-ND	FFC Conn	$0.60	$0.60	2	$1.20
WM11349-ND	FFC Jumper	$2.84	$2.84	1	$2.84
10118192-0001LF	MicroUSB Conn	$0.46	$0.46	1	$0.46
RMCF0805JT5K10	Resistor	$0.10	$0.00	4	$0.00
CL21B105KBFNNNG	Capacitor	$0.11	$0.00	1	$0.00
1-640456-4	14pos M Header	$0.88	$0.88	2	$1.76
1-640456-2	12pos M Header	$0.76	$0.76	1	$0.76
Total Cost = $66.77					

Table 4.1: Total Cost of Components

5 Conclusion

5.1 Ethics and Safety

With regards to the IEEE Code of Ethics I-1[9], There is a meager chance of injury caused by physical interaction such as snagging, as the product is relatively flat. There is a potential safety issue with a wearable electronic device powered by a LiPo rechargeable battery. To mitigate this risk, OSHA [10] recommends inspecting rechargeable batteries for damage or other signs of degradation. OSHA also recommends only using batteries that have been certified such as a UL2054 certified battery. The belt follows these OSHA-recommended practices when dealing with the battery. Our motors are self-contained coin type motors, meaning no rotating parts or pinch points will be exposed to the user.

Any overheat risk has been mitigated through proper power trace width of 10mils (or greater), as well as using 22-gauge insulation-coated for both signals and inter-board power. A 22-gauge solid copper core wire at 60 degrees Celsius has a current capacity of 3 amps[11]. The maximum current our BMS can supply is 1.5 Amps meaning that 22-gauge wire provides plenty of margin. Vents are also present on the casing containing the battery.

Sweat is a concern when creating a wearable exercise device. Excessive moisture may lead to damage to the system through short circuits or corrosion, and may even cause contacts to conduct to the user if touched with bare skin. To protect the device from sweat, all wires are insulation-coated, and all PCBs are enclosed in plastic casing. Our device is not intended to be used in the rain or other extreme weather conditions.

We do not anticipate our project to be accidentally or intentionally misused, and it does not collect any high-level or complex information about the user. The user would be informed of what the belt does, and what kind of information it collects.

5.2 Conclusions

In summary, our device will accurately measure cadence and provide haptic feedback to a runner. One of the major accomplishments of this project is the development of the step detection algorithm. This algorithm has proven to be very reliable and accurate. Another accomplishment is the switching circuit on the haptic feedback board was able to provide enough current through the vibrational motors to power them, along with the entire product being sufficiently powered by a single, small 3.7V LiPo battery.

However, A few uncertainties remain in the project. Of main concern is the MCU. While the microcontroller would program, we were never able to successfully run programs on the microcontroller board and instead had to use a development board. Our IMU would also not interface correctly and likewise forced us to use an IMU development board. If these two issues can be resolved, we have shown through the use of the dev boards that our project is viable and can achieve all of the high-level requirements previously outlined. Within the sphere of running, our project could have a moderate impact. Our target audience within the running community is relatively niche; professional runners and dedicated amateurs. The relatively low unit cost as well as simplicity of operation of our product could help increase adoption. These two categories of runners are the most outward-facing to the general public meaning our device could be thrust into the public eye if adopted by high-profile professional runners.

References

[1] "HUB," ARION. https://www.arion.run/hub/

[2] "Buy Stryd," Stryd (United States). https://www.stryd.com/us/en/store

[3] "BNO086," Digikey. https://www.digikey.com/en/products/detail/ceva-technologies-inc /BNO086/14114190

[4] "HXJNLDC 103048 Li-ion battery" Amazon. https://www.amazon.com/dp /B091XYZ2V3?ref=emc_s_m_5_i_n

[5] "ZiO LIPO Battery Manager (QWIIC, 3.7ViN, 0.5A/Charger, 3.3VoUt, 1.35Amax)," Smart Prototyping. https://www.smart-prototyping.com/Zio-LiPo-Battery-Manager.html

[6] "2N7002ET7G," Digikey. https://www.digikey.com/en/products/detail/onsemi /2N7002ET7G/13886993

[7] "Mini vibration motor 2.0mm," Seeed Studio, Sep. 26, 2023. https://www.seeedstudio.com /Mini-vibration-motor-2-0mm-p-2300.html/

[8] A. Industries, "Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055," www.adafruit.com. https://www.adafruit.com/product/2472

[9] IEEE, "IEEE Code of Ethics," ieee.org, Jun. 2020. https://www.ieee.org/about/corporate/governance/p7-8.html

[10] "Preventing fire and/or explosion injury from small and ...". OSHA. https://www.osha.gov/sites/default/files/publications/

[11] "Hook Up Wire — 3051/1 — Alpha Wire," www.alphawire.com. https://www.alphawire.com/products/wire/hook-up-wire/premium/3051_1/

Appendix A Microcontroller R&V Table

Requirement	Verification
The Microcontroller Board shall connect the supplied 3.3V power line from the BMS to the ESP32 MCU.	Equipment: DMM Procedure: 1. Set DMM to 1x DC Voltage. 2. Power system. 3. Attach positive probe to ESP32 3.3V Line. 4. Attach negative probe to ESP32 GND Line. 5. Record Voltage on DMM screen. Reporting: DMM measured voltage should be consistent with 3.3V +/- 0.1V tolerance.
The Microcontroller Board shall allow the ESP32 MCUto communicate on the HSPI_SCK (P14), HSPI_MISO (P12), and HSPI_MOSI (P13) pins, which shall be routed to the IMU Board.	Equipment: Debugging MicroUSB Cable, Laptop Procedure: 1. Power system. 2. Load SPI Testing program onto ESP32. 3. Jostle IMU Board, observe reported SPI metrics. Reporting: Testing program should receive SPI communication data consistent with desired output metrics per IMU datasheet.
The Microcontroller Board shall allow the ESP32 MCU to communicate on the GPIO25 (P25), GPIO26 (P26), GPIO32 (P32), and GPIO33 (P33) pins, which shall be routed to the Haptic Feedback Board.	Equipment: Debugging MicroUSB Cable, Laptop Procedure: 1. Power system. 2. Load GPIO Testing program onto ESP32 3. Actuate tactile switch on Haptic Feedback Board, observe reported metrics. 4. Observe vibration motors on Haptic Feedback Board, vibration pattern should be present. Reporting: Testing program should detect tactile switch actuation. Vibration motors should exhibit vibration pattern defined in testing program.

Appendix B Algorithm R&V Table

Requirement	Verification
The Algorithm shall accurately determine the User's Cadence in units of SPM.	Equipment: Computer with Arduino Software and Proper Environment, Metronome, MCU, IMU Procedure: 1. Program the MCU with the algorithm. 2. Open the Serial Monitor on the Computer. 3. Set the Metronome to a specific tempo. 4. Shake the IMU according to the tempo. Reporting: Serial Monitor should display cadence with a very close value to metronome tempo
The algorithm must respond according to the User's cadence in 3 scenarios: below 120SPM, between 120 SPM - 170SPM, and above 190 SPM.	Equipment: Computer with Arduino Software and Proper Environment, Metronome, MCU, IMU, Haptic Feedback Board Procedure: 1. Program the MCU with the algorithm. 2. Set the Metronome to 140SPM 3. Shake the IMU according to the tempo. 4. Shake the IMU extremely fast 5. Stop shaking the IMU Reporting: The haptic motors shall perform one long burst when IMU is shaken very rapidly, two small bursts when IMU is shaken at 140SPM, and nothing when the IMU is not shaken
The algorithm must adjust the target cadence according to 2 scenarios: decrease by 5SPM if the tactile switch is briefly pressed, and increase the target cadence by 5SPM when the tactile switch is pressed for longer than 2 seconds.	Equipment: Computer with Arduino Software and Proper Environment, MCU, Haptic Feedback Board Procedure: 1. Program the MCU with the algorithm. 2. Open the Serial Monitor on the Computer. 3. Briefly press the tactile switch three times 4. Press and hold the tactile switch for 3 seconds Reporting: Serial Monitor should display decrease in target cadence by 5 SPM when switch is briefly pressed, and an increase by 5 SPM when switch is pressed for 3 seconds

Appendix C IMU R&V Table

Requirement	Verification
The BMS shall supply up to 1.5 A at 3.3 +- 0.1V to the IMU Board, Microcontroller Board, and Haptic Feedback Board.	Equipment: DMM Procedure: 1. Set DMM to 1x DC Voltage. 2. Power system. 3. Attach positive probe to BMS 3.3V Line. 4. Attach negative probe to BMS GND Line. 5. Record Voltage on DMM screen. 6. Set DMM to 1x DC Current. 7. Jostle IMU Board to simulate steps. 7. Record Current on DMM Screen Reporting: DMM measured voltage should be consistent with 3.3V +/- 0.1V tolerance. DMM measured current should be below 1.5A tolerance.
The BNO086 IMU shall supply Linear Acceleration Vectors, Absolute Orientation Vectors, and other metrics to the Microcontroller Board through a SPI interface.	Equipment: Debugging MicroUSB Cable Laptop w/ Arduino Procedure: 1. Power system. 2. Load SPI Testing program onto ESP32. 3. Jostle IMU Board, observe reported SPI metrics. Reporting: Testing program should receive SPI communication data consistent with desired output metrics per IMU datasheet.

Appendix D Haptic Feedback R&V Table

Requirement	Verification
The Haptic Feedback Board shall connect the supplied 3.3V power line from the BMS to the 2N7002ET7G MOSFET and the PTS526 SM15 SMTR2 LFS tactile switch.	Equipment: DMM Procedure: 1. Set DMM to 1x DC Voltage. 2. Power system. 3. Attach positive probe to ESP32 3.3V Line. 4. Attach negative probe to ESP32 GND Line. 5. Record Voltage on DMM screen. Reporting: DMM measured voltage should be consistent with 3.3V +/- 0.1V tolerance.
The PTS526 SM15 SMTR2 LFS tactile switch shall connect to the Microcontroller Board via signals connected to the GPI pins of the ESP32 MCU.	Equipment: Debugging MicroUSB Cable, Laptop Procedure: 1. Power system. 2. Load GPIO Testing program onto ESP32. 3. Actuate tactile switch on Haptic Feedback Board, observe reported metrics. 4. Observe vibration motors on Haptic Feedback Board, vibration pattern should be present. Reporting: Testing program should detect tactile switch actuation. Vibration motors should exhibit vibration patter defined in testing program.
The 2N7002ET7G MOSFET shall provide the supplied 3.3V power line to both SEEED STUDIO 2.0MM MINI vibration motors when sufficient voltage is applied to its gate via the GPO signal from the Microcontroller Board.	Equipment: DMM Procedure: 1. Set DMM to 1x DC Voltage. 2. Power system. 3. Attach positive probe to motor positive Line. 4. Attach negative probe to motor negative Line. 5. Record Voltage on DMM screen. Reporting: DMM measured voltage should be consistent with 3.3V +/- 0.1V tolerance.