

VARIABLE SPEED RUNNING MUSIC DEVICE

by

Nithin Boopalan

Weston Heard

Alan Wan

Design Document for ECE 445, Senior Design, Fall 2026

TA: Jianchong Chen

Project No. 40

1. Introduction

1.1 Problem and Solution:

Most people bring their phone on a run to listen to music. It is big and bulky, and unless you have a perfect playlist, no music syncs with your running, and even pace matched music does nothing to enforce that you are running at that speed. This matters because the runner's cadence is one of the few parameters that a runner can change, also just happening to be one of the most consequential. A study done by Anderson et al. found that increasing step rate reduces loads at many of the runner's joints, including the ankle, knee, and hip, while decreasing step rate increases them [1]. Running is among the most common forms of exercise while also being very high in injury rate. So helping recreational runners hold a higher and steadier cadence can improve their personal health and safety.

Van Dyck et al. found that runners held the most consistent cadence when the beat was matched to their own cadence [2]. Matching music to cadence works, but only when this match is maintained. Static playlists cannot do that, as a runner's pace can speed up or slow down over a run, while a track's tempo does not. Once the runner's cadence and musical tempo drift apart, the music starts to compete with the runner's stride instead of supporting it. The alternatives all carry costs: formal cadence retraining needs an exercise lab or a coach, phone applications require carrying a phone, which is never comfortable, and running watches only report cadence after a run. The device will clip to the runner's waistband, where an IMU measures the cadence. The device then plays the music over a pair of wired earbuds. There are two modes, the Follow mode where the music gets

WSOLA time stretched and transitioned to match tempo with the runner's pace, while keeping the same pitch. The other mode is Target, where it attempts to keep the runner at a certain pace by applying filters when off pace to let you know to slow down or to speed up. All of this will be done without the hassle of having to bring a phone with you.

1.2 Visual Aid



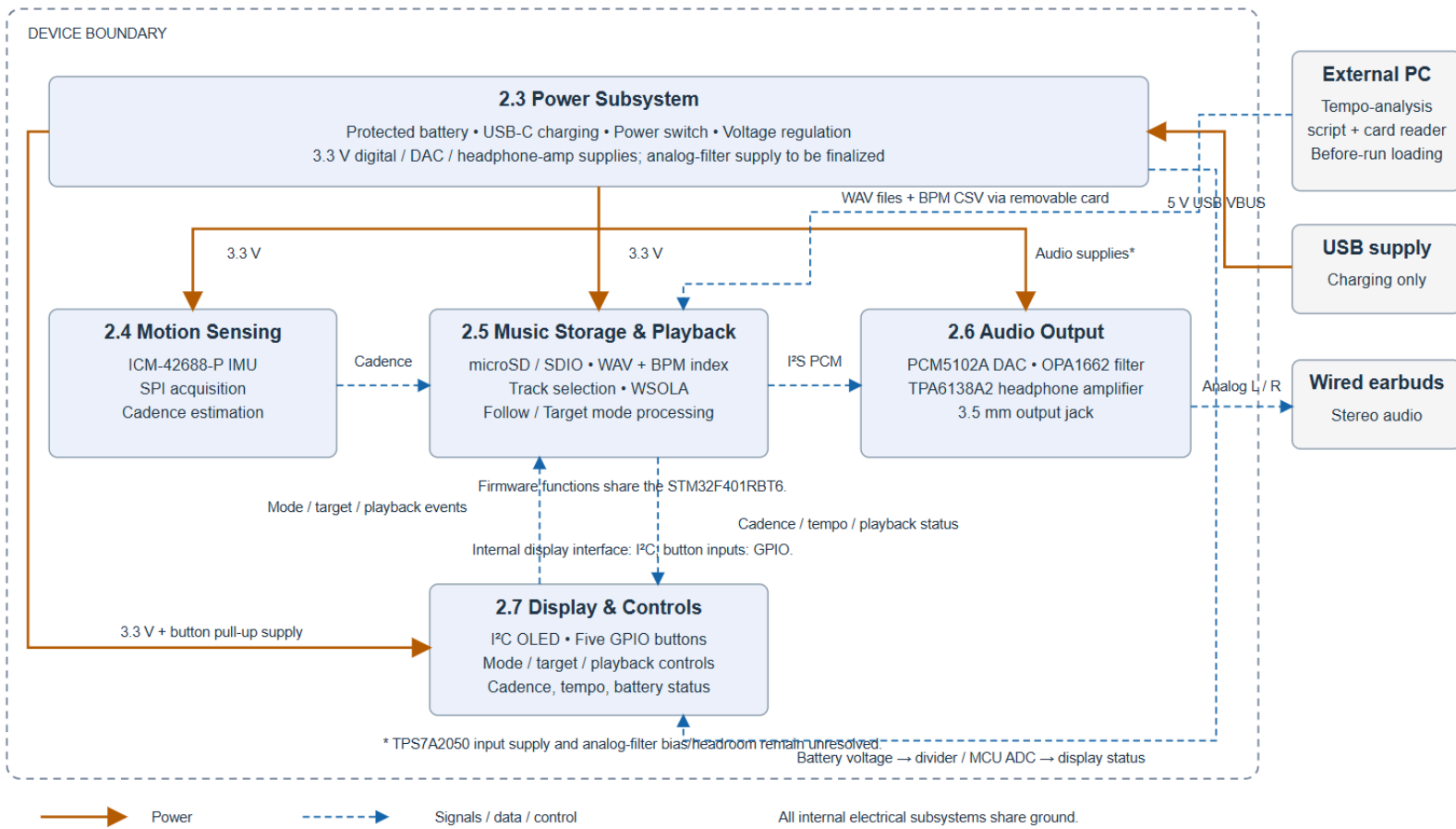
The device clips on to the user's waist. It has two ports, a Usb-C (for charging and flashing the chip) and a 3.5mm headphone jack. The songs are loaded on through a computer onto a microSD card, where they are analyzed and tagged with their beatmatching. The device turns on with a switch on the side. The volume is controlled with the up and down arrows, and the middle button toggling playback, and if held changes the current listening mode. The side arrows change the bpm in target mode, and change songs in follow mode.

1.3 High-level requirements list:

- The device must measure the runner's cadence to within 1 step per minute of the true value across the running range of 140 to 190 steps per minute, so that the tempo selected is an accurate reflection of the runner's stride.
- The device must also hold the playback tempo to within 1 beat per minute of the measured cadence (or exactly half or double), and converge to a new tempo within 10 seconds of a pace change of at least 15 steps per minute
- The device must operate on its own for at least 2 hours of continuous use, without any external device at any point.

2. Design

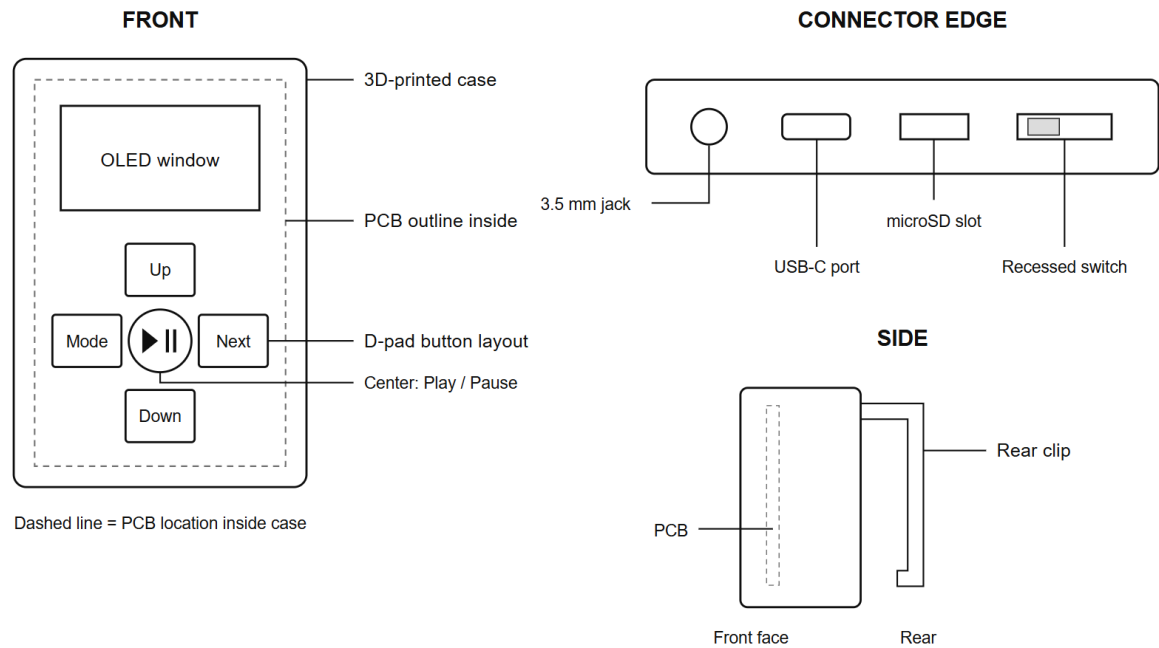
2.1 Block Diagram:



The device consists of power, motion sensing, music storage and playback, audio output, and display/control subsystems. The battery supplies the device through a recessed power switch and a buck-boost convertor configured for 3.3V. The USB-C will provide charging power to the subsystem. The ICM-42688-p communicates with the STM32F401RBT6 over SPI, and the firmware estimates running cadence from the motion data. A microSD card supplies audio and a tempo index over SDIO. Based off of this the MCU selects tracks and performs pitch preserving temp adjustment when it is in Follow Mode, or maintains a user-selected tempo and applies digital feedback filtering when in Target Mode. I²C carries the processed audio to the PCM5102A, followed by the analog filter, headphone amp,

and the 3.5 mm jack. The OLED displays operating status, 5 GPIO buttons provide controls, and an ADC input monitors the battery voltage.

2.2 Physical Design (if applicable):



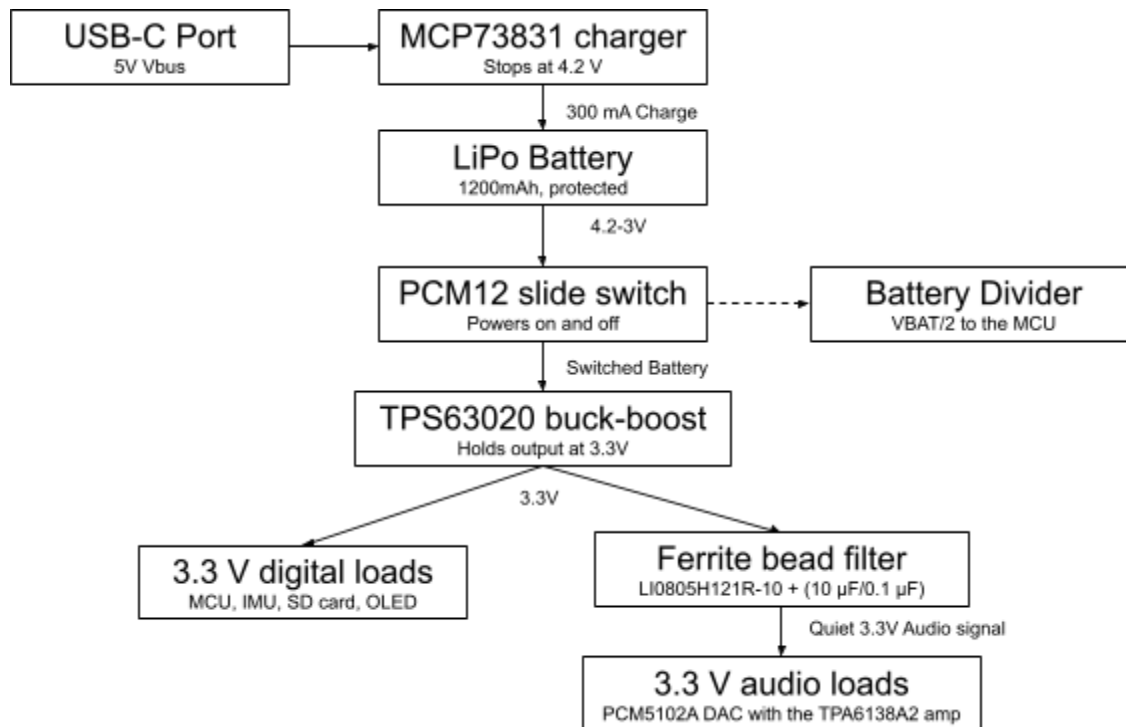
This is a diagram of our enclosure that will surround all the internal components of our device. The front face will contain a window for the OLED screen to show through. It will also have the 5 buttons built into it which will be above the actual components on the PCB. On the edge there will be 4 openings for the 4 ports that we will be using: USB-C, microSD, 3.5mm jack, and lastly just a power switch. The PCB will sit on the inside of the device and will be connected by screw mounts which will also be printed in the casing. The final part of the physical design will be a clip on the back of the case to attach to the users waistband.

SUBSYSTEMS:

2.3 Power Subsystem

The power subsystem is what supplies all of the other blocks from a rechargeable LiPo cell, and is what will allow us to hit our high level requirement of at least 2 hours of standalone operation. It recharges the cell over USB-C. The subsystem consists of:

- protected 1200 mAh LiPo cell
- PCM12SMTR slide switch
- TPS63020 buck-boost converter that produces the regulated 3.3 V rail
- ferrite-bead filter which produces a separate quiet 3.3 V audio rail
- MCP73831 charge controller fed from USB-C
- resistor divider so the microcontroller can measure battery voltage.



We chose a single-cell 3.7 V LiPo (Adafruit #258, 1200 mAh) because the charger and converter are both designed for one lithium cell, and a flat pouch cell fits a waistband clip. The cell has a built-in protection circuit that disconnects it on overcharge, over-discharge and short circuit. It connects through a keyed JST-PH connector that prevents reverse insertion, which also makes it easy to replace.

The PCM12SMTR slide switch is placed between the battery and the rest of the system, after the charger connection. That means the device can still charge while switched off, and doesn't draw from the battery while stored.

The cell voltage falls from 4.2 V to about 3.0 V during discharge, which crosses the 3.3 V that our circuit needs. A linear regulator would drop out below about 3.4 V and waste the remaining capacity. The TPS63020 buck-boost steps the voltage down when the battery is above 3.3 V, and steps the voltage up when it is down. It accepts 1.8-5.5 V in, where the output voltage is set by the feedback divider:

$$\frac{R_1}{R_2} = \frac{V_{OUT}}{V_{FB}} - 1 = \frac{3.3}{0.5} - 1 = 5.6$$

If we choose R2 to be 100kΩ, then we would get an R1 of 560kΩ

$$V_{OUT} = 0.5V \left(1 + \frac{R_1}{R_2} \right) = 0.5V \left(1 + \frac{560k\Omega}{100k\Omega} \right) = 3.30V$$

For the audio rail, the DAC and headphone amplifier run from a separate 3.3 V rail. An LI0805H121R-10 ferrite bead with 10 μ F and 0.1 μ F capacitors isolates it from the digital rail. This filter keeps the converter's switching noise and the digital current spikes out of the audio signal.

The MCP73831 charges the cell from USB-C at 300 mA and stops at 4.2 V. A resistor divider lets the microcontroller read the battery voltage for the on-screen battery status. The firmware will shut down the device once it hits 3.3V to protect the battery cell.

Component	Typical Current
STM32F401RBT6 microcontroller	20mA
ICM-42688-P IMU	1mA
Streaming from microSD card	30mA
OLED display	15mA
PCM5102A DAC	20mA
TPA6138A2 amplifier + headphones	16mA
Total	102 mA

For estimate purposes, if we give ourselves a 1.5 times margin, at a 90% converter efficiency with the average 3.7V of the battery, we get:

$$I_{BAT} = \frac{(3.3V)(102mA)(1.5)}{(0.9)(3.7V)} = 152mA$$

If we estimate that only 75% of the battery capacity is usable, then we get:

$$t = \frac{0.75(1200mAh)}{152mA} = 5.9 \text{ hours}$$

Requirements	Verification
The 3.3 V rail must stay between 3.135 and 3.465 V ($\pm 5\%$) for battery voltages from 3.0 to 4.2 V and loads up to 300 mA	Connect bench power supply in place of the battery and an electronic load from Test Point 3V3 to Ground Set the supply to 3.0, 3.7 and 4.2 V, and at each set the load to 0, 150 and 300 mA Measure Test Point 3V3 with a multimeter and record the results to confirm all values are in range
With USB-C connected, the battery must charge and stop at 4.158 - 4.242 V ($4.2 \text{ V} \pm 1\%$)	Start with a partially discharged battery Connect USB-C and monitor the battery voltage with a multimeter until charging stops Record the final voltage and confirm it's in range
The device must shut down when the battery reaches $3.30 \pm 0.05 \text{ V}$	Connect a bench supply in place of battery, starting at 3.6V with the device running. Lower the voltage in 10mV steps. Record the voltage when the device shuts off.
The device must play music for at least 2 hours from a full battery charge (satisfies high level requirement)	Fully charge the battery Start Follow-mode playback at maximum volume into 32 Ω headphones Record the time until the device shuts down. Confirm it is at least 2 hours

2.4 Motion Sensing Subsystem

This subsystem determines the runner's cadence in steps per minute, the core measurement the rest of the device depends on. An IMU mounted at the hip will sample acceleration at a rate of 200 Hz and pass the data to the STM32 over SPI, batched through the sensor's internal FIFO so that the processor only needs to read groups of samples rather than needing to service an

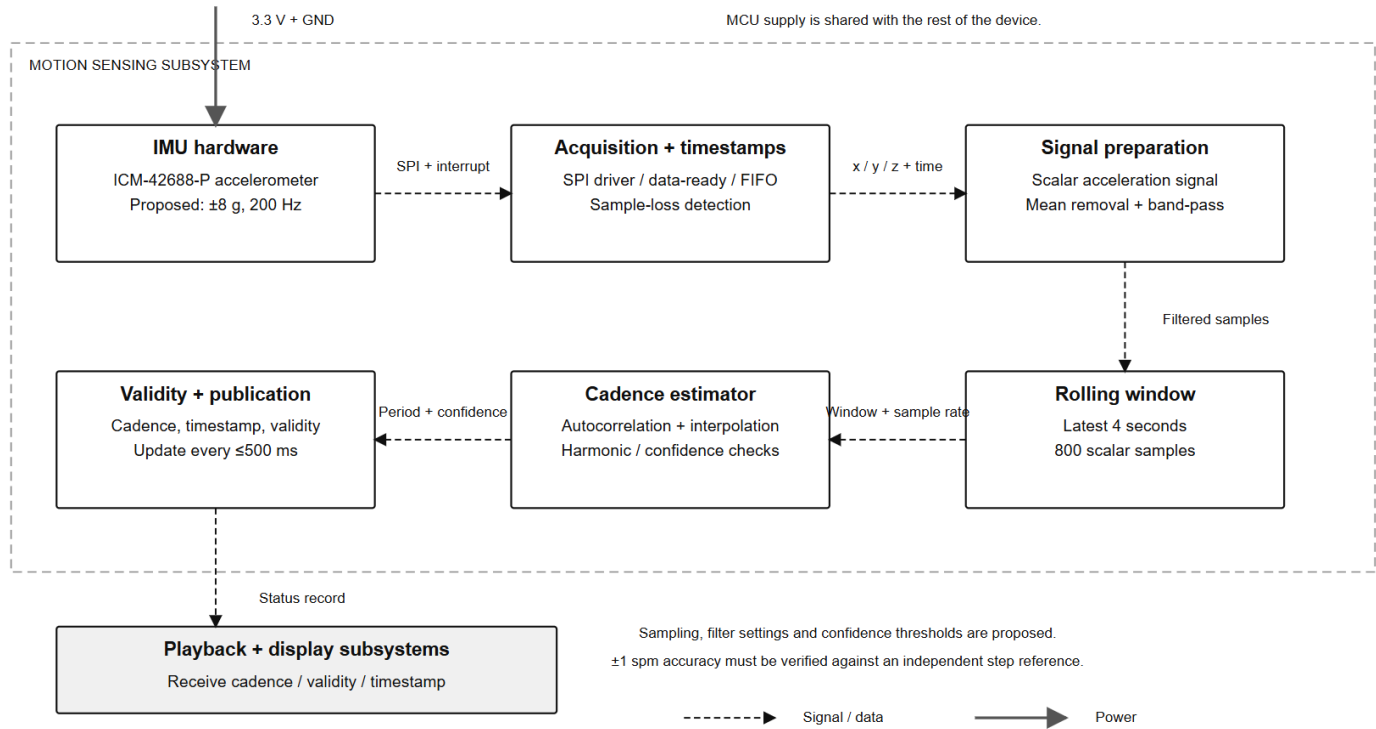
interrupt for each sample. The firmware then isolates the gravity vector from the raw readings by applying a low-pass filter and projecting the acceleration onto it, so the measurement isn't affected by the angle at which the device is worn. That signal is then passed through a band-pass filter to the running and walking range to remove the gravitational offset and vibration/noise the sensor may pick up.

The cadence itself is found through autocorrelation, where the firmware compares a four-second window of the filtered signal against shifted copies of itself. The shift that best matches is the stride period, as it indicates the length of one stride. This is preferred over counting the individual footsteps since, with an entire window, a single missed step or peak barely moves the result. This shift range first covers the running range of 140 to 190 spm, then widens to the walking range, done in steps to not confuse a runner at 180 spm with a walker at 90 spm.

One thing to note is that because the shift can only be measured in a whole number of samples, the raw search can only report certain cadences at discrete values. To fix this issue, the firmware will need to estimate a fractional shift rather than the integer result. This can be done by interpolating the values from the autocorrelation function as a parabola, where the vertex contains the true peak. The mathematical showcase will be performed in the 2.8 tolerance

analysis section.

Hardware + firmware blocks; firmware runs on the shared STM32F401RBT6.



Requirements	Verification
High-level requirement 1: Measured cadence must be within ± 1 spm of true cadence from 140 to 190 spm.	Have a user test the device on a treadmill at a set speed and compare results to make sure cadence stays within ± 1 spm of true cadence from 3 different paces between 140 to 190 spm.
The IMU initializes correctly and supplies all three acceleration axes at the configured rate without invalid SPI transactions.	Read the identity/configuration registers, apply known orientation changes, and capture decoded SPI traffic for ten minutes. Verify axis response and configuration.
After a controlled change of ≥ 15 spm within 140–190 spm, cadence reaches and remains	Replay time-stamped IMU recordings with known step timing and confirm worn trials

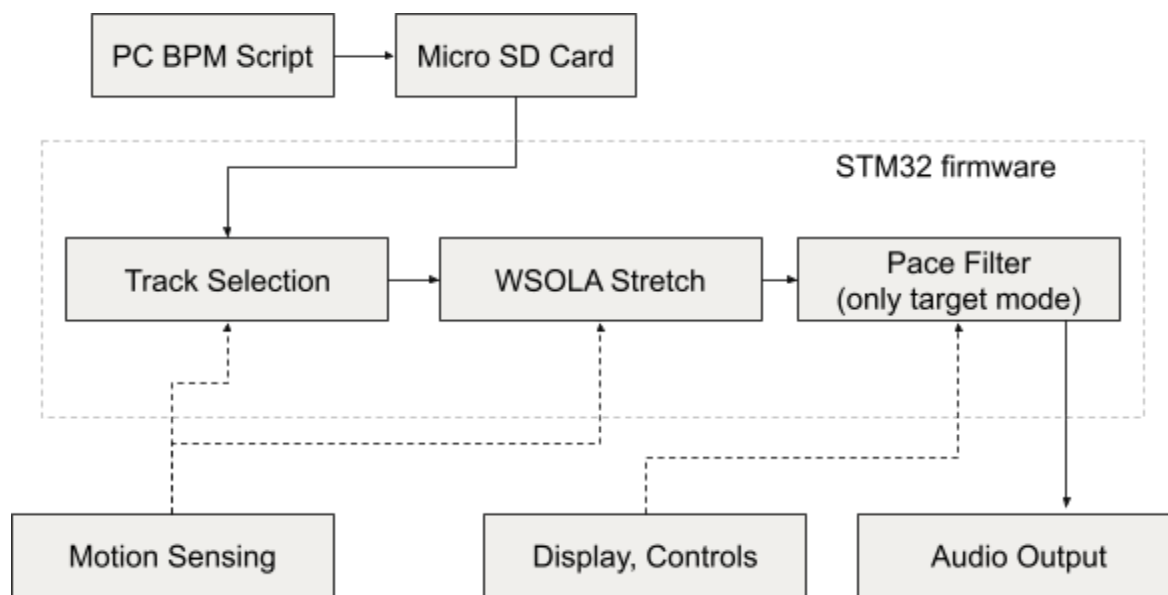
within ± 1 spm of the new reference within six seconds.	where the reference actually changes.
No missing-data or clipped-data window is reported as valid; stopping acquisition produces an invalid output within one second.	Inject missing samples and clipped values into the acquisition pipeline and separately disable sensor communication. Check that affected windows are invalid, faults are logged, and loss is announced within one second.

2.5 Music Storage and Playback Subsystem

Music is prepared on a computer before a run. A script analyzes each track's tempo and writes the results to a CSV index file stored alongside the audio on the SD card. The device then only performs a lookup at runtime rather than having to analyze audio in real time. The device then selects a track from its library whose tempo is in the range of the measured cadence and applies WSOLA time-scale modification to adjust the playback tempo without altering the pitch of the song. Small drifts in pace can be fixed by adjusting the stretch used; a pace change too large to alter cleanly will cause the device to switch to a different track instead.

The device will have 2 modes. In follow mode, the music continuously tracks the runner so the beat stays locked in without the runner needing to think about it. In target mode, the runner sets an ideal cadence, and the device holds the music at that tempo. If the runner's cadence is too slow or fast, the device will apply a low-pass filter on the music that progressively muffles the audio and again slowly clears up as they return to it. This gives the runner continuous feedback on their pace without requiring any input on the runner's side.

Audio is stored as 16-bit, 44.1 kHz stereo WAV files, requiring 176.4 kB/s from the microSD card over SDIO. Track selection picks a song whose BPM is within 10% of the measured cadence. WSOLA then stretches it by a ratio between 0.90 and 1.10, and if a larger change is needed the device switches tracks. In Target mode, no filtering is applied within ± 3 spm of the target; beyond that, the low-pass cutoff decreases until it reaches its strongest effect at ± 15 spm off target.



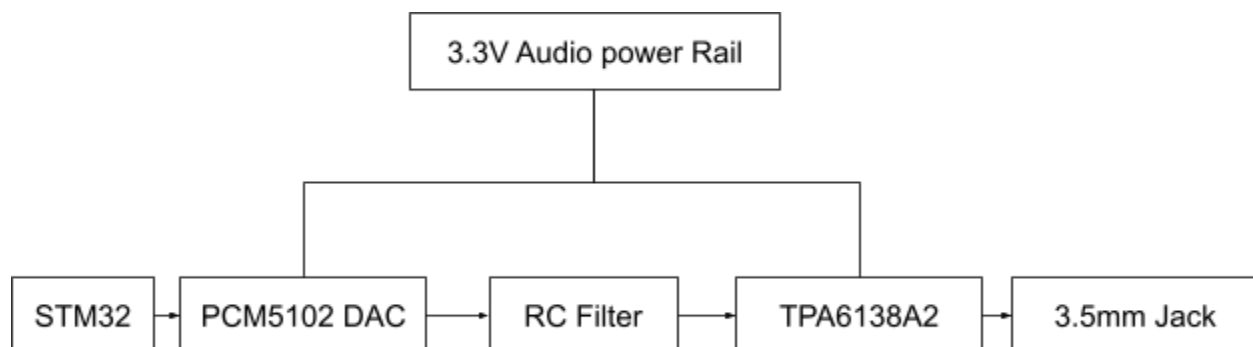
Requirements	Verification
High-level requirement 2: Playback tempo must be within ± 1 BPM of the measured cadence (or half or double based on the tracks original BPM)	Feed fixed cadences of 150, 170 and 190 spm to the firmware over UART Log track (BPM * stretch ratio) over UART Confirm each result is within ± 1 BPM of the cadence
High-level requirement 2: Tempo must converge within 10s after a 15 spm change.	Step the simulated cadence from 160 to 175 spm over UART Time how long until the logged tempo is within ± 1 BPM of 175.

	Confirm ≤ 10 s.
Stretch ratio must stay between 0.90 and 1.10; beyond that, the device must switch tracks.	Sweep the simulated cadence from 140 to 190 spm Log stretch ratio and track ID over UART Confirm the ratio never leaves 0.90-1.10 and the track changes when it should
Playback must have zero audio dropouts over 10 minutes	Play music continuously for 10 minutes Read the firmware's buffer underrun counter over UART and confirm it is 0

2.6 Audio Output Subsystem

The audio output subsystem converts the digital audio from the micro controller into an analog signal that can be used for 16-32 Ω wired earbuds. It outputs the final tempo-matched music that our second high-level requirement depends on, and it sets a hardware limit on maximum volume to protect the user's hearing. The subsystem consists of four parts which run from the filtered 3.3V audio rail:

- PCM5102A DAC,
- RC output filter
- TPA6138A2 headphone amplifier
- 3.5 mm headphone jack



The STM32 sends 16-bit stereo audio at 44.1kHz to the PCM5102A.. We chose the PCM5102A because it creates its own internal clock and produces a 2.1V RMS output from a single 3.3V supply. Following its datasheet, a 470 Ω /2.2nF RC filter on each output removes most of the high-frequency noise.

The TPA6138A2 drives the earbuds directly. Its gain is set by two resistors so that even at maximum volume the output cannot exceed 150mV RMS, the EN 50332 limit for portable music players:

$$|G| = \frac{R_F}{R_{IN}} = \frac{2.2k\Omega}{33k\Omega} = 0.067 \rightarrow V_{max\ OUT} = 2.1V(0.067) = 140mV\ RMS$$

The user can adjust the volume digitally in firmware.

Requirements	Verification
The output can't exceed 150 mV RMS into a 32 Ω load at max volume	Connect a 32 Ω resistor across the headphone jack output Play a full-scale 1kHz test tone at max volume Measure the RMS voltage on each channel with an oscilloscope and record values
Frequency response must be flat within ± 1 dB from 20Hz to 20kHz	With the 32 Ω load, play test tones at 20Hz, 100Hz, 1kHz, 10kHz and 20kHz at a fixed volume Measure the amplitude of each with an oscilloscope Record the results in a table and confirm all are within ± 1 dB of the 1kHz value
DC offset at headphone jack must be within ± 5 mV with no audio playing	With the 32 Ω load connected and nothing playing, measure the DC voltage on each channel with a multimeter, record values

2.7 Display and Controls Subsystem

The display will allow the runner to select which mode, set a target cadence, control music playback, view the cadence, and view the device's battery status. It supports the device's autonomous operation requirement needed for a run without having to use a phone. It also exposes the measured cadence and selected playback tempo so that the user can identify the current operating state. Display work will have lower priority than audio-buffer servicing and cadence acquisition, ensuring that interaction does not prevent the device from meeting its tempo-tracking requirements.

The proposed display is a 0.96-inch, 128×64 monochrome OLED module using an SSD1306 controller. The controller supports this resolution and I²C communication. A complete module explicitly rated for a 3.3 V supply and 3.3 V logic will be selected.. The OLED requires no separate backlight and can display the necessary status information in a small enclosure. Actual current consumption will be measured at the selected contrast setting and included in the two-hour battery budget.

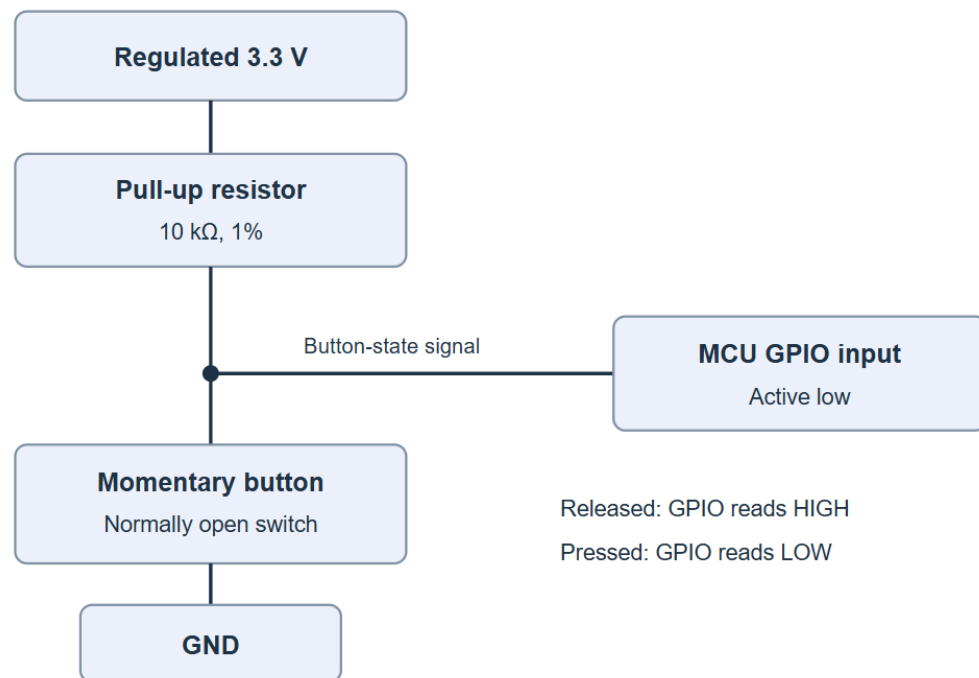
Sending the data through the display will require a framebuffer which will require:

$$128 \times 64 \text{ pixels} \times 1 \text{ bit/pixel} / 8 = 1,024 \text{ bytes.}$$

The driver will send 32-byte data chunks. Each chunk contains an address byte, control byte, and 32 data chunks. From this we calculated our nominal transfer time below:

$$t_{\text{chunk}} = (1 + 1 + 32) \times 9 / 400,000 = 0.765 \text{ ms.}$$

The buttons that will control the display will be a normally open momentary switch connected between an MCU GPIO and ground. Therefore when you press, the input will be the active low drawing the current. The firmware will sample the buttons every 5 ms and accept a state change after four consecutive identical samples, giving a nominal 20 ms debounce interval. Each debounced press will create one event. This is the circuit design we aim for below:

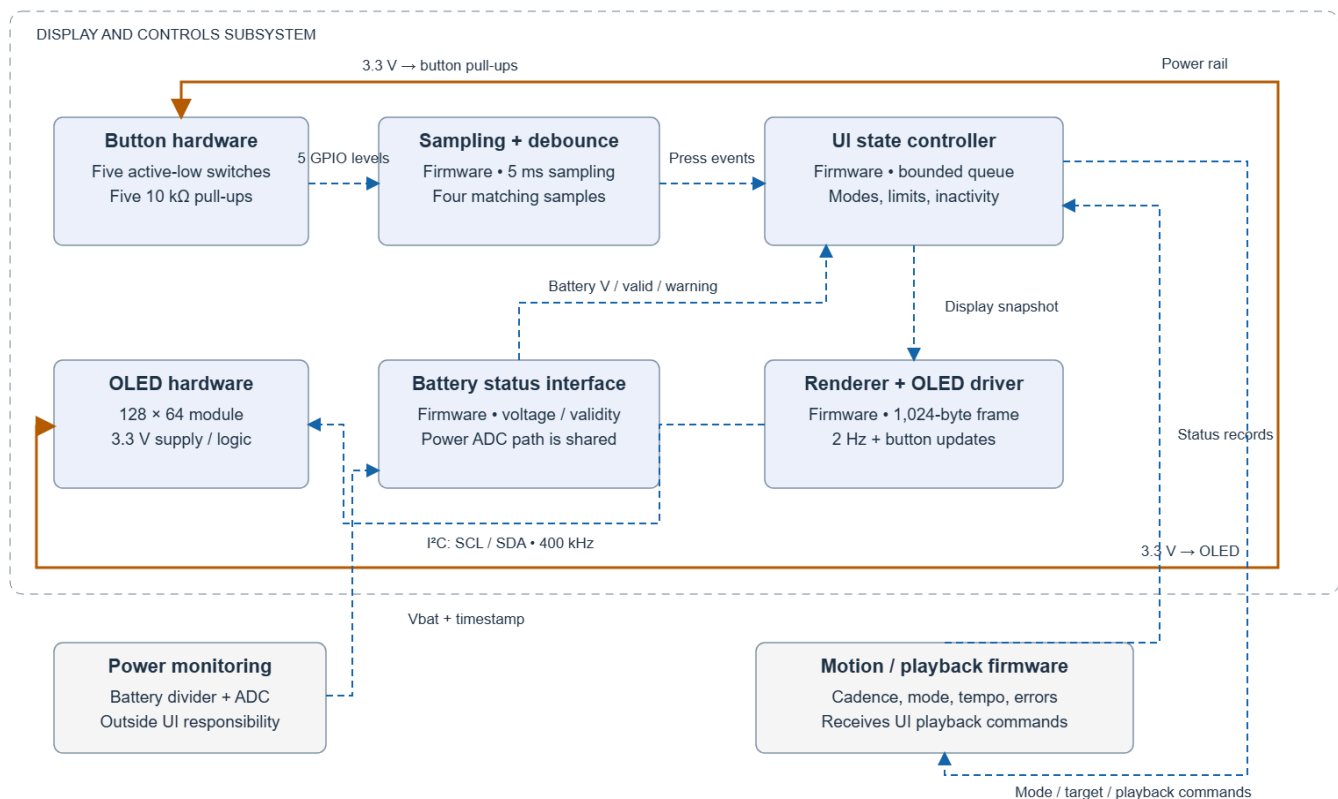


These will be the buttons that we are using:

1. Play/Pause/Mode: Toggle playback, pausing retains the selected track and selected mode. Holding down this button for more than two seconds will switch the mode.
2. Right: In Target mode, increase target cadence by 1 step/min, up to 190. In Follow mode, increase digital volume by one of 16 discrete levels.

3. Left: In Target mode, decrease target cadence by 1 step/min, up to 140. In Follow mode, decrease digital volume by one level
4. Up: Request the next available track
5. Back: Go back to the last track

Subsystem Block Diagram:



Requirements and Verifications for this Subsystem:

Requirements:	Verifications:
The selected OLED module and controls function at the right voltage level without an external operating device.	Disconnect any USB power and operate on battery alone. If all of the required functions operate properly then it works.
Released buttons read high, pressed buttons low, and each held-button pull-up current is ≤ 0.36 mA at the maximum specified supply.	Measure each GPIO voltage and resistor current to confirm any high/low readings and current is within limit
The device supports all user operations for at least 2 hours.	Have a user test the device on a run and track the total time until battery depletion.
Target settings span 140-190 spm in steps of exactly 1 spm; volume spans levels 0-15 without exceeding any limits.	Sweep both settings through all values to make sure it covers all and only the required values.

2.8 Tolerance Analysis:

2.8.1 Source of the Cadence Error

Let $x[n]$ be the filtered acceleration signal where n indexes the samples at frequency f_s .

Cadence can be obtained from the autocorrelation of the signal, written as $r[k]$, which is a measure of how closely the signal resembles a copy of itself.

$$R[k] = \sum (x[n] \cdot x[n+k])$$

The shift k that maximizes $r[k]$ is our one stride period.

$$\text{cadence} = 60 * f_s / k$$

This, however, only evaluates at integer values of k , and thus only certain cadences are ever reachable. Since cadence is inversely proportional to k , the spacing between the reachable values is not uniform. As the cadence increases, the spacing also grows. At 190 spm with our $f_s = 200$.

$$k = 60 * 200 / 190 = 63.158$$

Our 2 closest neighbors are 63 and 64 samples.

$$60 * 200 / 63 = 190.476 \text{ spm}$$

$$60 * 200 / 64 = 187.5 \text{ spm}$$

$$190.476 - 187.5 \approx 3 \text{ spm spacing}$$

$$3 / 2 = 1.5 \text{ spm worst-case error}$$

2.8.2 Fractional Lag Interpolation

The firmware therefore needs to estimate a fractional lag from the shape of the correlation's peak. Near a smooth vertex, the first derivative is zero, and the second-order term dominates, meaning $r[k]$ is locally parabolic around the ideal k value. Letting k be the integer lag that won the initial search, the difference in k and the true peak can be defined as a function of the terms closest to k .

$$\delta = 0.5 \cdot (r[k-1] - r[k+1]) / (r[k-1] - 2 \cdot r[k] + r[k+1])$$

$$k_{\text{true}} = k + \delta$$

2.8.3 True Error

The accuracy of the estimate is now limited by two things: the Model bias since the peak is not perfectly a parabola, and the noise present in our signal.

Model Bias Error:

Model the normalized peak as $r[k+x] = \cos(\omega(x-\epsilon))$, with $\epsilon \in [-0.5, 0.5]$

$$r[k-1] - r[k+1] = -2 \cdot \sin(\omega) \cdot \sin(\omega\epsilon)$$

$$r[k-1] - 2 \cdot r[k] + r[k+1] = 2 \cdot \cos(\omega\epsilon) \cdot (\cos \omega - 1)$$

$$\delta = 0.5 \cdot (r[k-1] - r[k+1]) / (r[k-1] - 2 \cdot r[k] + r[k+1])$$

$$= 0.5 \cdot \cot(\omega/2) \cdot \tan(\omega\epsilon)$$

$$e_{\text{bias}}(\epsilon) = \delta - \epsilon = (\omega^2/12) \cdot \epsilon \cdot (4\epsilon^2 - 1)$$

Maximizing over ε gives $\varepsilon = 1/\sqrt{12} = 0.2887$, where $\varepsilon(4\varepsilon^2-1) = -0.1925$:

$$|e_{\text{bias}}| \leq (9.896 \times 10^{-3}/12) \cdot 0.1925 = 1.59 \times 10^{-4} \text{ samples}$$

Noise Error:

Near the peak, δ is sensitive to the two outer values and almost insensitive to the center one:

$$\partial\delta/\partial r[k-1] = +0.5/D, \quad \partial\delta/\partial r[k+1] = -0.5/D, \quad \partial\delta/\partial r[k] \approx 0$$

$$\sigma_{\delta} = \sqrt{0.5^2 + 0.5^2} \cdot \sigma_r/\omega^2 = 0.707 \cdot \sigma_r/\omega^2$$

At $\varepsilon = \pm 0.5$

$$\sigma_{\delta} \leq 1.5 \cdot \sigma_r/\omega^2$$

$$\sigma_w = 70 \times 10^{-6} \cdot 9.81 \cdot \sqrt{7.3} = 1.855 \times 10^{-3} \text{ m/s}^2$$

$$\eta = 1.855 \times 10^{-3} / 5 = 3.71 \times 10^{-4}$$

$$\sigma_r = 2 \cdot 3.71 \times 10^{-4} / \sqrt{800} = 2.62 \times 10^{-5}$$

$$\sigma_{\delta} = 1.5 \cdot 2.62 \times 10^{-5} / 9.896 \times 10^{-3} = 3.97 \times 10^{-3} \text{ samples}$$

Taking 3 standard deviations

$$|e_{\text{noise}}| \leq 3.97 \times 10^{-3} \cdot 3 = 1.19 \times 10^{-2} \text{ samples}$$

$$e_{\text{total}} = 1.59 \times 10^{-4} + 1.19 \times 10^{-2} = 1.21 \times 10^{-2} \text{ samples}$$

$$e_{\text{bound}} = 0.0121 \text{ samples}$$

$$\text{error} = 0.0121 \cdot S = 0.0121 \cdot 3.008 = 0.036 \text{ spm}$$

This maximum combined error of 0.036 is now under our specification requirements of 1 spm.

3. Cost and Schedule

3.1 Cost Analysis:

3.1.1 Labor:

Based on the ECE graduate salaries, assuming 40 hours a week and 52 weeks per year, the hourly rate is \$42/hour. Each member is budgeted 150 hours, divided as shown in the table below. $\text{Hours} \times 2.5 \times \text{the hourly rate}$ gives each member's labor cost.

Partner	Design/Implementation Hours	Integration/Testing Hours	Documentation Hours	Total Cost
Weston	90	40	20	\$15,750
Nithin	90	40	20	\$15,750
Alan	80	50	20	\$15,750

3.1.2 Parts:

Description	Manufacturer	Part #	Qty	Source	Cost
Microcontroller, Cortex-M4, 84 MHz	STMicroelectronics	STM32F401RBT6	1	E-shop	\$0.00
8 MHz crystal		Crystal – 8.0000 MHz 8 pF	1	E-shop	\$0.00
USB-C receptacle, SMD	GCT	USB4105-GF-A-060	1	E-shop	\$0.00
Tactile push button	C&K	PTS645SL43SM TR92	5	E-shop	\$0.00
Ferrite bead, 120 Ω @ 100 MHz	Laird	LI0805H121R-10	1	E-shop	\$0.00
microSD card, 32 GB	Samsung	MB-MP32DA	1	ECE Supply Center	\$13.05

6-axis IMU, SPI	TDK InvenSense	ICM-42688-P	1	Digi-Key	\$5.50
Stereo audio DAC, I ² S	Texas Instruments	PCM5102APWR	1	Digi-Key	\$4.60
Headphone amplifier	Texas Instruments	TPA6138A2PWR	1	Digi-Key	\$1.90
Buck-boost converter	Texas Instruments	TPS63020DSJR	1	Digi-Key	\$4.30
1.5 μ H power inductor	Coilcraft	XFL4020-152ME	1	Digi-Key	\$1.20
Li-ion charge controller	Microchip	MCP73831T-2A CI/OT	1	Digi-Key	\$0.70
LiPo battery, 3.7 V 1200 mAh	Adafruit	258	1	Digi-Key	\$9.95
JST-PH 2-pin connector	JST	S2B-PH-SM4-TB	1	Digi-Key	\$0.50
Slide switch, SPDT	C&K	PCM12SMTR	1	Digi-Key	\$0.70
microSD card socket	Hirose	DM3AT-SF-PEJ M5	1	Digi-Key	\$2.50
3.5 mm stereo jack, SMD	CUI Devices	SJ-3524-SMT-TR	1	Digi-Key	\$1.20
0.96" 128×64 OLED module, I ² C	Generic (SSD1306)		1	Amazon	\$6.99
Subtotal					\$53.09

3.1.3 Sum of costs into a grand total

Item	Cost
Parts subtotal	\$53.09
Shipping (5%)	\$2.65

Sales tax (10%)	\$5.31
Total parts	\$61.05
Labor (3 * \$15,750)	\$47,250.00
Grand total	\$47,311.05

3.2 Schedule:

Week	Course milestone	Weston (Hardware)	Alan (Firmware)	Nithin (Integration)
9/28	Design Document due 10/2	Select architecture. Create power and audio schematics	Select cadence algorithm and tempo method	Select OLED, buttons, SD card format (WAV)
10/5	Design Review, PCB Round 2 due 10/8	Finish schematic and PCB layout, order PCB	Read IMU over SPI on dev board, test autocorrelation on data	Buy parts, SD card read and WAV playback on dev board
10/12	Breadboard Demo, PCB Round 3 due 10/15	Breadboard DAC -> filter -> amp chain, change PCB if needed	Cadence detection working on dev	Start PC BPM analysis script, OLED and button

			board, WSOLA prototype on PC	drivers on dev board
10/19		Assemble PCB power subsystem, then MCU verify 3.3 powerrails	Port WSOLA to MCU	PC script writes BPM CSV, design enclosure
10/26	PCB Round 4 due 10/29	Assemble audio chain, IMU and SD socket, final board touchup/fixes	Cadence firmware running on PCB	Port SD, OLED and buttons to PCB
11/2	Progress Demo, progress reports due 11/4	Audio output tests (max output level, frequency response)	Follow mode end to end, begin Target mode low-pass	Print enclosure v1, integrate firmware modules
11/9	Team Contract assessment due 11/13	2-hour battery life test	Track switching at the $\pm 10\%$ stretch limit	Integrate prototype into enclosure, refine fit

11/16	Mock Demo and Mock Presentation	Refine prototype: hardware fixes	Refine prototype, tune cadence filter and response time	Test integrated system, outdoor run tests, R&V checks
11/23	Fall Break			
11/30	Final Demo and Presentation video due 12/4	Final R&V verification (power and audio)	Final R&V verification (cadence and tempo)	Final R&V verification (display, controls, enclosure)
12/7	Final paper due 12/9, lab checkout 12/10	Final paper hardware sections	Final paper firmware sections	Final paper integration and testing sections

4. Discussion of Societal Impact, Engineering Standards, Ethics, and Safety

Considerations:

4.1 Discuss how your project makes a positive contribution to public health, safety and welfare considering economic, environmental, social, cultural, and global factors.

Our device helps recreational runners (who are most likely not professionals) keep a higher and steadier cadence, which can reduce load to the ankles, knees, and hips which may reduce injuries. Normally to train cadence, one would need a coach, or an expensive device to track them, so by making it cheaper it makes it more affordable to keep a safer pace. It works without a phone, doesn't need internet, and doesn't need a subscription, so it can be given as a gift or a loan and can continue to be used to its full potential, anywhere as long as it is charged. Also, with a rechargeable battery, that means that there is no waste created like with disposable batteries.

4.2 Identify the engineering standards (IEEE, ACM, UL, etc.) that apply to your project.

Our design follows the following standards:

EN 50332: limits the maximum output of portable music players to 150 mV, which corresponds to about 100 dBA in typical headphones. Our amplifier gain is set to stay under this limit [4].

IEC 62133-2: safety requirements for portable lithium cells. We use a protected cell and a charger with a fixed 4.2 V limit [5].

USB Type-C specification: governs how the device draws 5 V charging power, using 5.1 k Ω resistors on the CC pins [6].

IPC-2221: general PCB design rules for trace width and spacing [7].

4.3 Explain how IEEE/ACM codes of ethics and course ethics guidelines relate to your project. Discuss ethical concerns for your project.

The IEEE Code of Ethics requires us engineers "to hold paramount the safety, health, and welfare of the public" [3]. Our main ethical concern is hearing. When listening to loud sounds for a prolonged period of time, like with a pair of earbuds, a music device can cause permanent hearing damage. That is why we cap the output volume in hardware, not just in software, so no setting can exceed the EN 50332 limit. A second concern is situational awareness, since some runners wearing earbuds may not hear traffic. We will recommend moderate volume in our documentation, however do not want to completely limit it as some may not be using it in an area with vehicles (trails, mountains, backyard, etc.). In keeping with the Code's requirement for honesty, we will not claim that the device prevents injuries, it only helps runners maintain a cadence. The device has no wireless connection and collects no personal data, so it poses no privacy risk.

4.4 Discuss electrical and mechanical safety concerns referring to course safety guidelines page. Add any ethical and safety concerns that arose since your proposal.

The main electrical hazard is the lithium-polymer cell, which can swell, vent or catch fire if it is overcharged, over-discharged, short-circuited or physically damaged. The device runs below 5 V, so there is no shock hazard. Mechanically, the device is worn at the hip during exercise which incites a lot of motion, so like our TA discussed with us, our enclosure must have no sharp edges. Since it is moving we also want to make sure that the earbud cable doesn't snag. To combat some of this, we have moved the power switch after the charger, replaced the fuel gauge with a simpler voltage divider, and corrected our volume limit to the EN 50332 standard (150 mV, about 100 dBA).

4.5 Document procedures to mitigate the safety concerns of your project

Charging: following the course battery safety guidelines [8], cells are charged only while someone is present, on a non-flammable surface, and stored in the lab's designated battery area.

Inspection: any swollen or damaged cell is removed from use.

Hardware protections: We are making sure the cell's built-in protection circuit cuts it off on overcharge, over-discharge and short circuit, the MCP73831 limits charging to 300 mA and 4.2V, the keyed JST connector prevents reverse insertion, and firmware shuts the device down at 3.3V.

Hearing: The amplifier's fixed gain caps the output at 140 mV RMS. The amplifier's built-in short-circuit protection guards against a shorted plug.

Mechanical: the enclosure fully covers the battery and board and has rounded edges, to make sure the battery does not pose a risk.

Testing: running tests will be done on a track, treadmill, or pedestrian trail, away from traffic.

5. Citations:

[1] R. W. Anderson, C. Bonanno, H. Menz, S. Bade, and M. Lawrenson, "What is the effect of changing running step rate on injury, performance and biomechanics? A systematic review and meta-analysis," *Sports Medicine - Open*, vol. 8, no. 1, 2022.

- [2] E. Van Dyck, B. Moens, J. Buhmann, M. Demey, E. Coorevits, S. Dalla Bella, and M. Leman, "The power of auditory-motor synchronization in sports: Enhancing running performance by coupling cadence with the right beats," PLOS ONE, vol. 8, no. 8, 2013.
- [3] IEEE, "IEEE Code of Ethics," IEEE Policies, Section 7.8, 2020. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>
- [4] Sound System Equipment – Headphones and Earphones Associated with Portable Audio Equipment – Maximum Sound Pressure Level Measurement Methodology, EN 50332-1.
- [5] Secondary Cells and Batteries Containing Alkaline or Other Non-Acid Electrolytes – Safety Requirements for Portable Sealed Secondary Lithium Cells, IEC 62133-2.
- [6] USB Implementers Forum, Universal Serial Bus Type-C Cable and Connector Specification. [Online]. Available: <https://www.usb.org/usb-type-cr-cable-and-connector-specification>
- [7] IPC, Generic Standard on Printed Board Design, IPC-2221.

[8] ECE 445 Senior Design Laboratory, "General Battery Safety," University of Illinois Urbana-Champaign. [Online]. Available:
<https://courses.grainger.illinois.edu/ece445/documents/GeneralBatterySafety.pdf>