

BUDGET CLIP-ON POSTURE CHECKER

ECE 445 DESIGN DOCUMENT - FALL 2025

Team #33

Yue Cao, Ashit Anandkumar, Edward Ruan, Destiny Jefferson

Professor: Arne Fliflet

TA. Wenjing Song

Contents

1 Introduction

1.1 Problem.....	3
1.2 Solution.....	3
1.3 Visual Aid.....	4
1.4 High-Level Requirements.....	5

2 Design

2.1 Block Design	6
2.2 Functional Overview & Requirements	7
2.3 Tolerance Analysis	15

3 Costs and Schedule

3.1 Cost Analysis.....	17
3.2 Schedule.....	19

4 Ethics and Safety

4.1 Ethics and Safety.....	20
----------------------------	----

5 Citations

5.1 Citations.....	22
--------------------	----

1 Introduction

1.1 Problem

More than 80 percent of jobs in the United States involve mostly sedentary activities, resulting in long sitting hours [1]. When one sits for extended periods of time, it is quite common for their posture to change, with that change often being slouching, which can result in chronic back and shoulder pain, fatigue, and respiratory issues such as trouble breathing. Of course, sedentary lifestyles apply to more than just jobs. For example, sitting for extended periods of time while playing video games on a computer can lead to a slouched posture.

Current solutions for this issue involve a tight brace, which can be restrictive and uncomfortable for users [2]. Some solutions are also very impractical, such as one that uses a camera that the user needs to sit in front of in order to track their posture [3].

1.2 Solution

The Clip-On Posture Checker is a small wearable clip-on device that monitors your posture and provides real-time feedback via a vibration motor, LED, or buzzer to indicate when your posture has deviated too far from the desired position. The device not only clips onto the user's shirt but also has shoulder straps for additional security. There will be a calibration feature in the form of a button that, when held down, sets the current position of the sensors as the "desired position."

1.3 Visual Aid

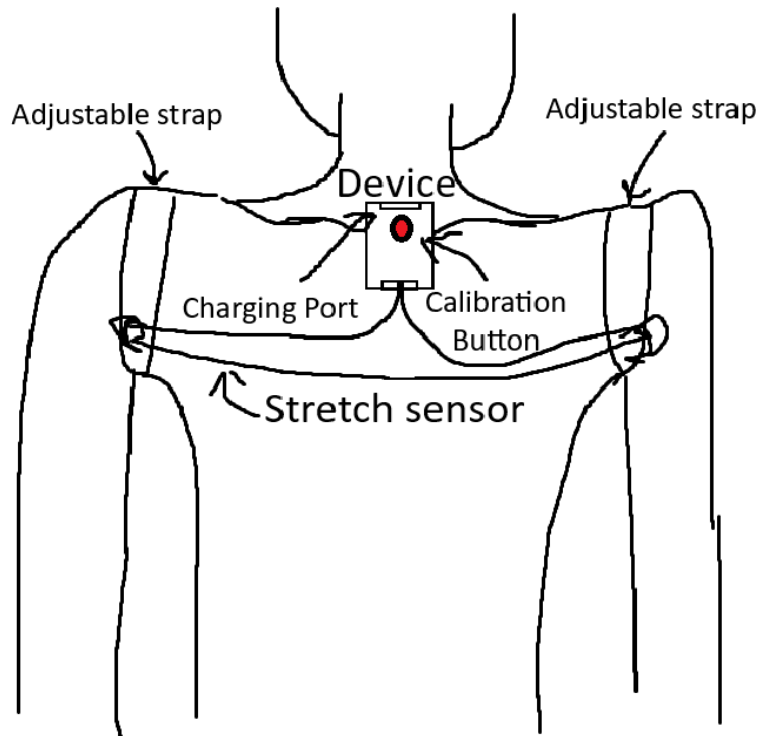


Figure 1: Visual of what the Budget Clip-On Posture Checker will look like.

The device will feature a calibration button (represented by a red circle in Figure 1) that enables the user to set their desired “proper posture.” The device will take roughly 2 seconds to calibrate before the sensors (most of which are enclosed inside of the device) regularly check the user’s posture to see if it has deviated too far from what is desired. The adjustable straps allow the device to be used by those with both wide and narrow shoulders, allowing the device to be used by a wide range of people. The stretch sensor acts as an additional measure of posture and will track if the user is slouching their shoulders.

1.4 High-Level Requirements

- I. The device shall detect the user's torso tilt angle relative to the calibrated upright posture with an accuracy of at least $\pm 5^\circ$.
- II. The device shall provide real-time feedback (vibration or LED) within 3 seconds of when posture deviation exceeds a threshold angle (e.g., 15° forward lean).
- III. The stretch sensor attached to the clip-on device will act as a secondary point of measurement to ensure accurate measurement of posture deviation from calibration. It will detect elongation of the shoulders corresponding to poor posture with $\pm 3\text{mm}$ change in length, to help distinguish between breathing and posture deviations.

2 Design

2.1 Block Diagram

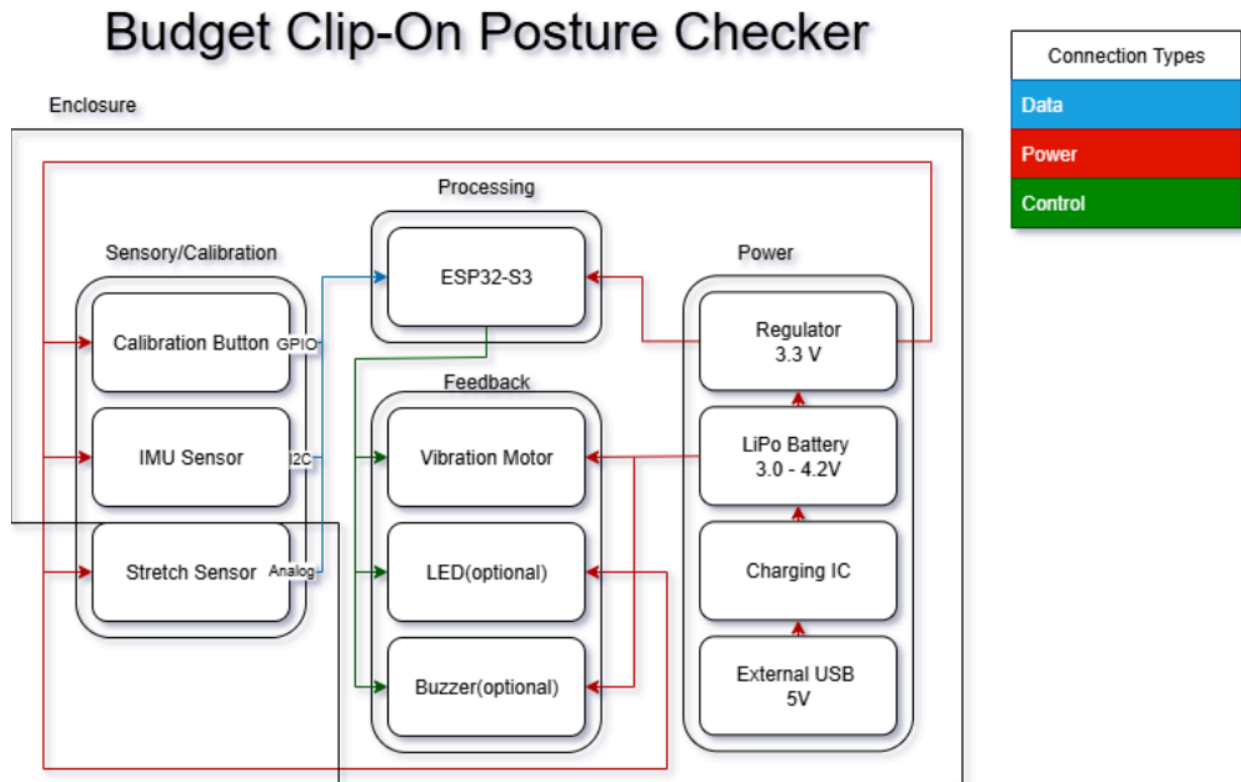


Figure 2: Block Diagram

2.2 Functional Overview & Requirements

2.2.1 Sensory Subsystem

The sensory subsystem goal is to measure the user's body orientation and also the back curvature in relation to the user's calibrated position. This subsystem will contain an IMU (ICM-20948), a resistive stretch sensor and a calibration button. This Subsystem will be connected to the Processing subsystem as the data collected in the system will need to be processed by the Processing system. The IMU that we chose to utilize in this project is the ICM-20948 as this contains both an accelerometer and a gyroscope which will be crucial in detecting the user body orientation and determining whether it deviates from the calibrated position. The stretch sensor we will use is 12/14" Flexible Stretch Sensor, this will help detect the curvature of the back and will help detect any slouching in the user's body orientation and is crucial in detecting the deviation from the calibrated position. This will be done by using a simple voltage divider circuit to measure the change in voltage as the sensor stretches. The user will be required to wear adjustable shoulder straps with anchors for the stretch sensor to attach to. The calibration button is used to be able to capture the users desired position and be used in calculating deviation.

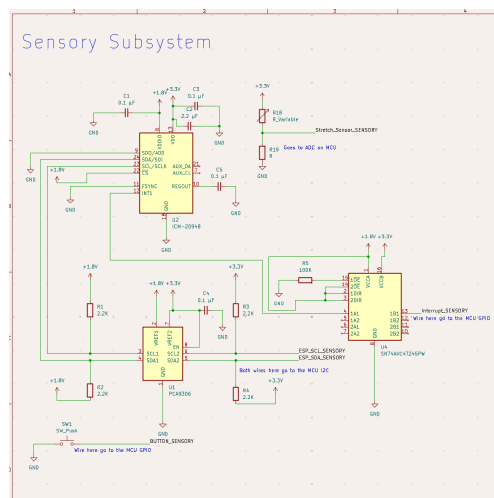


Figure 3: KiCad Schematic of the sensor subsystem

Requirements	Verification
The IMU shall provide accelerometer and gyroscope readings at ≥ 50 Hz which results in 50 readings taken per second	• Run a 10-second loop that prints the sample readings • Count total samples after 10 s and ensure that that the samples counted ≥ 500 • Store these samples into an excel sheet to save the data
The system shall detect a change in stretch of ≥ 3 mm across the shoulders averaged over 10 seconds	• Mount the strap on a stable setup • From neutral, extend to 3 mm and hold for 10s • Log ADC values during the hold • Ensure that the feedback system is alerted and an alert is provided after 10 seconds • Store these samples into an excel sheet to save the data
The signal will be set at 1.65 V ± 0.05 to be able to detect a slouch	• With stretch sensor set at neutral, measure the ADC input using a DMM ensuring it is within 1.60 - 1.70 V • Briefly apply a small stretch and show that the reading decreases from the set position and can capture a slouch • Take a photo of the DMM at the neutral position and when the sensor is deviated
The calibration button should only detect a single press per time which will be done through a debouncer with 20ms	• Probe the button GPIO with an oscilloscope • Press the button 10 times at normal speed • Read the MCU's press counter using a serial output • And ensure that the MCU only detected 10 presses not more or not less • Store these samples into an excel sheet to save the data

Table 1: Requirements and verifications for the Sensor subsystem

2.2.2 Feedback Subsystem

The goal of the Feedback subsystem is to provide real-time feedback to the user in order for them to 1) recognize that their posture has deviated from the ideal posture and 2) be able to rectify that issue by adjusting their posture to their original calibrated position. This subsystem will contain, at a minimum, a DC vibration motor that has an operating range between 2V and 3.6V [4], but we are also planning to include an LED and buzzer [5] that will be additional methods of notifying the user of any posture deviations. This subsystem will be connected to the Processing subsystem, which is where we will determine if the motor, buzzer, or LED needs to be triggered.

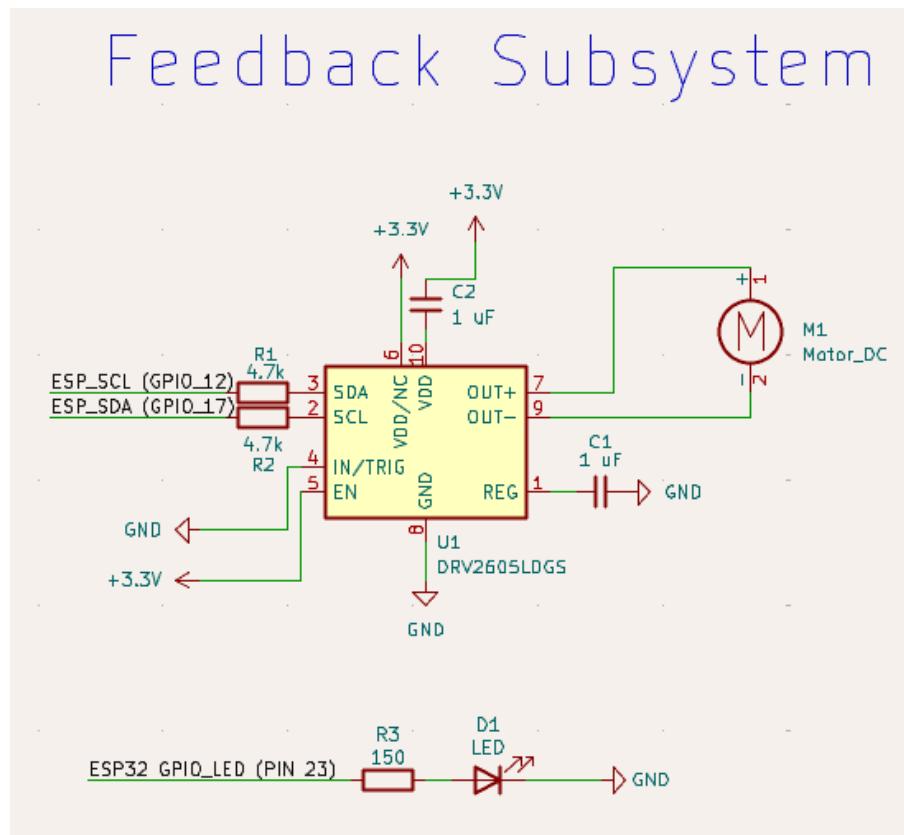


Figure 4: KiCAD Schematic of Feedback Subsystem

Requirements	Verifications
The LED, buzzer, and/or vibration motor are activated after a deviation of the user's posture has been detected, with the deviation occurring over an average of 10 seconds.	● Mount the strap on a stable setup ● From neutral, extend to 3 mm and hold for 10s ● Ensure that the Feedback system is alerted, and an alert is provided after 10 seconds ● Once the Feedback subsystem is alerted, the vibration motor should vibrate, the buzzer should beep, and the LED should light up.
The LED, buzzer, and/or vibration motor should all turn off after a previous posture deviation has been corrected and the correct posture is maintained for at least 2 seconds.	● Follow the previous requirement steps to ensure that the LED, buzzer, and motor are currently active ● Have the user adjust posture to neutral and hold for 2-4 seconds ● After 2-4 seconds of corrected posture, the LED, buzzer, and motor should all turn off.

Table 2: Requirements and Verifications for Feedback Subsystem

2.2.3 Power Subsystem

The Power Subsystem provides the power to all the other components of the device and is essential for the device to function. Without the power subsystem, the entire device will not function at all as no components will have power. Referring to the figure below, the first stage of the power subsystem is the USB4085-GF-A(USB Type-C connector) which allows for external input to charge the battery and power the system through a USB-C connection. This external charging connection feeds this power to the MCP73812T-420I/OT which is a Li-ion charging IC. This charging IC allows for a safe and reliable charging to the next component in the power subsystem, the LIPO801735 400mAh 3.7V Li-ion polymer battery which will serve as the source of power for when the device is not being charged. This battery powers the rest of the subsystems through the voltage-regulator stage consisting of the TLV75533PDBVR and TLV7551BPDBV. The TLV75533PDBVR will regulate the voltage down to 3.3 V for operations of the microcontroller, sensor subsystem, and feed back system, the TLV7551BPDBV will regulate the said 3.3 V down to 1.8 V for parts of the sensor subsystem that handles the data collection. To ensure proper power delivery to all subsystems, calculations can be referred to in the Tolerance Analysis section.

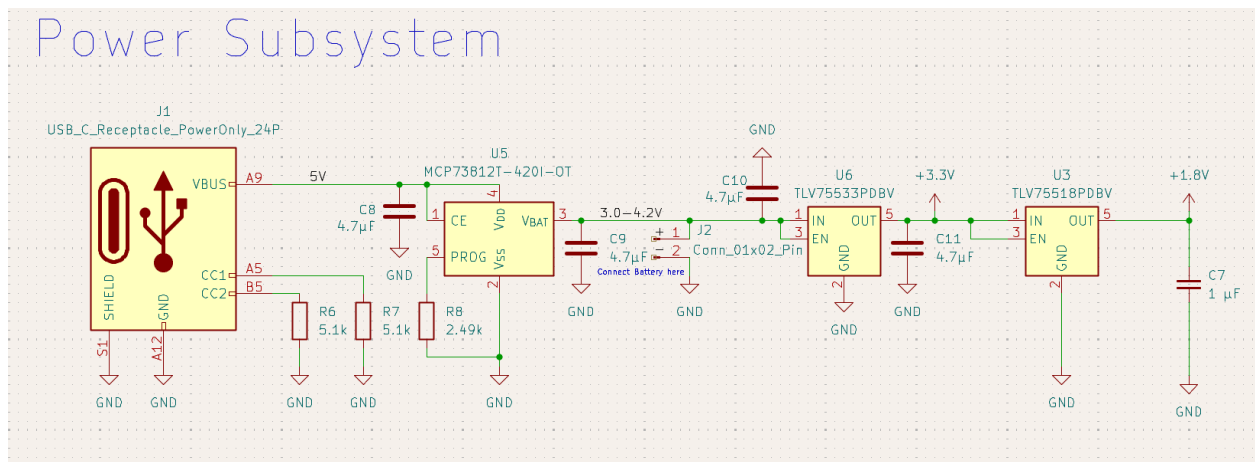


Figure 4: KiCad Schematic of the power subsystem

Requirements	Verification
The device must remain operational for at least 8 hours continuously on a fully charged 400 mAh 3.7 V Li-Po battery.	● Fully charge the battery using the MCP73812T-420I/OT IC, then disconnect from external power. Power the device on and record the start/end time of when the system shuts off automatically. ● Measure the average current draw during operation by probing battery pins with a multimeter or current meter, confirm that $420 \text{ mAh} / I_{\text{avg}} \geq 8 \text{ hours}$. ● Repeat this test on three separate occasions to ensure consistency of results in a $\pm 5\%$ consistency.
The MCP73812T-420I/OT charging IC should charge the battery from 3.0 V to 4.2 V within 2.5 hours, when powered through the USB4085-GF-A	● Connect external power via the USB-C connector, measure the battery voltage vs. time during a charging cycle, ensuring that $4.20 \pm 0.05 \text{ V}$ is reached within 2.5 hours. ● Probe the PROG pin using a multimeter or current meter, measure the charging current at the start and confirm the charging current tapers $< 10 \text{ mA}$ at full charge ● After the current drops below 10 mA, the charger must remain in standby mode so no further charging pulses should be observed.
The TLV75533PDBVR voltage regulator output should maintain	● Probe the TLV75533PDBVR output port, measure the output voltage using an

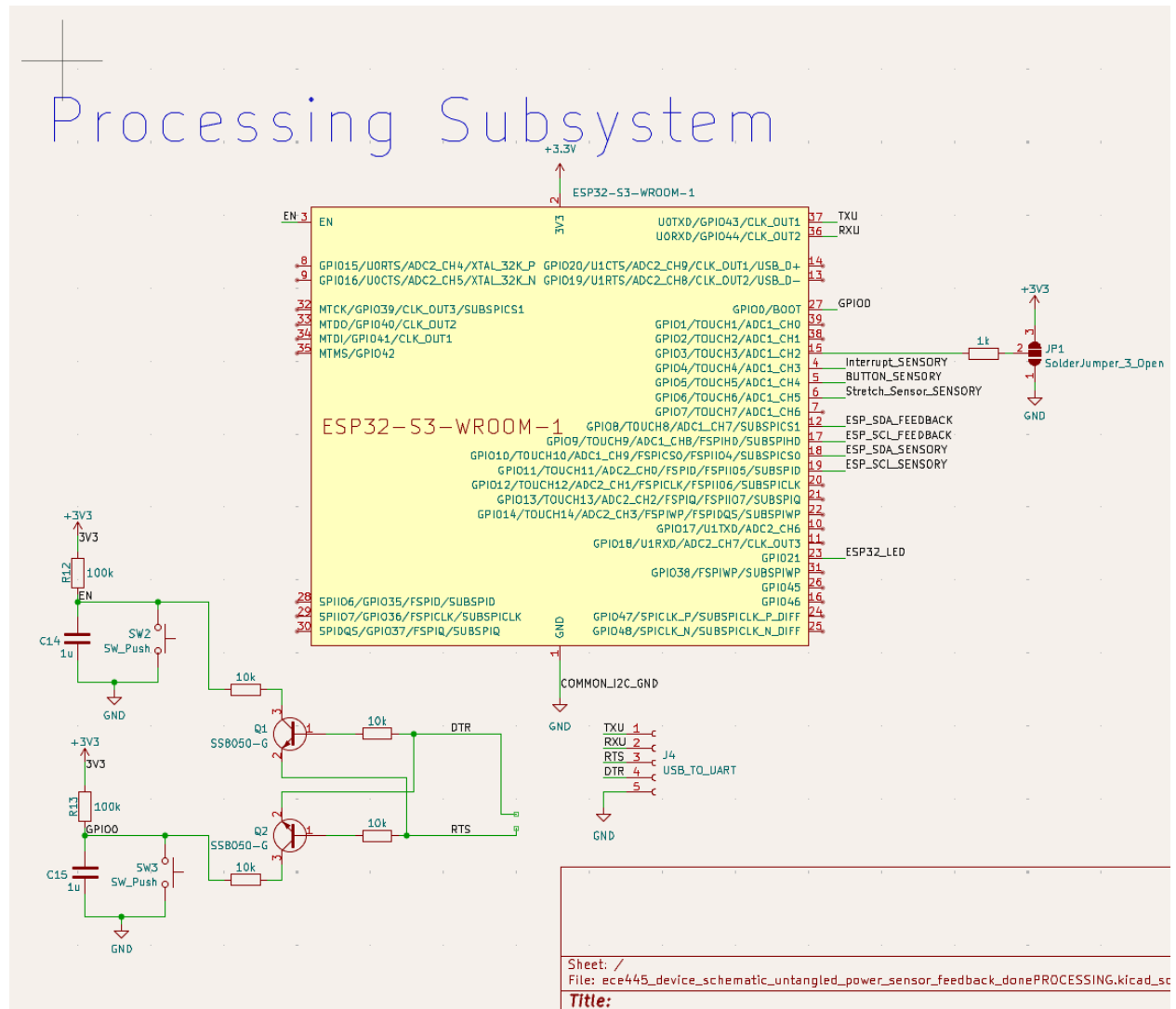
a steady 3.3 V $\pm$ 5% under all charging and operating conditions.	oscilloscope and verify that the readings stay within 3.135 - 3.465 V. • Ensure maximum load is applied by turning on the device, stretching the sensor, and actively try to trigger the feedback system, using the same procedure, measure the output voltage and confirm it does not fall below 3.135 V. • Now plug in the external USB-C connection to begin charging, monitor the regulator output with the oscilloscope to detect that the readings stay within 3.135 - 3.465 V.
--	---

Table 3: Requirements and verifications for the power subsystem

2.2.4 Processing Subsystem

The Processing Subsystem is centered on the ESP32-S3-WROOM-1 microcontroller module because it has basic GPIOs and common interfaces like I2C and is readily available in ESP32 Dev Kit. It serves as the central controller for the posture checker, interfacing with the IMU sensor(via I2C), stretch sensor(via ADC pins), haptic feedback driver (via I2C), external programming source(via a USB-to-UART bridge). It receives a 3.3V input from the power system. It collects sensor data, sensor interrupts and sensor calibration events via I2C and ADC,

performs posture analysis using embedded algorithms, determines whether the user is in poor posture and issues vibration feedback based on computed results.



Requirements	Verifications
The processing system auto-enters programming mode when buttons aren't pushed, and the USB-to-UART bridge is connected to an active computer USB output and auto-enters normal boot mode otherwise.	1. Connect the MCU to the PC through the USB-to-UART bridge. 2. Test all four cases (USB active/inactive × buttons pressed/released). 3. for each case, record EN/GPIO0 timing and UART messages.

	Pass if and only if the computer reports stub running on Arduino IDE for the “USB active + buttons released” case and normal firmware banner for all others.
The MCU should read the IMU and stretch sensor at ≥ 50 samples/second and should calculate the posture readings at ≥ 20 times/second	1. Load firmware that timestamps each IMU sensor data, stretch sensor data, and compute events. 2. Run for 120 s while recording counters and I2C traffic. Verification passes if every second window shows ≥ 50 samples for each sensor and ≥ 20 computation results.
The MCU should record the calibrated posture after the calibration button is pressed for 2 seconds	1. Press button for 0.5 s, 1.5 s, 1.9 s, 2.0 s, 2.1 s, 3.0 s (3 times for each). 2. Observe firmware “calibrated posture saved” message, timestamp, and the relevant metrics. Pass if no calibration occurs for < 1.9 s presses and events always occurs at 2.0 ± 0.1 s.
The MCU should activate the feedback subsystem via I2C when it detects a torso tilt of $\geq 15^\circ$ for 10 seconds from the calibrated position or when the stretch change is $>$ than the threshold(counting from the moment that MCU receives first poor posture data) and deactivate when the user is back to good posture for more than 4 seconds.	1. Calibrate at 0°, then tilt to 15° and hold 12 s while recording angle and I²C command time. 2. Return upright and measure deactivation delay with printed decision logs. 3. Repeat using stretch input above threshold for 3 s then below threshold. Verification passes if activation occurs 10.0 ± 0.2 s after the first poor sample and deactivation occurs 4.0 ± 0.2 s after good posture in all trials.
I2C bus voltage levels must remain within 0–3.3 V with logic HIGH ≥ 2.3 V and LOW $\leq$	1. Capture I2C waveforms during active communication and record VHIGH and

1.0 V.	VLOW. 2. Verification passes if for both SDA and SCL lines $V_{HIGH} \geq 2.3$ V, $V_{LOW} \leq 1.0$ V
GPIO output logic HIGH shall be ≥ 2.4 V and $LOW \leq 0.4$ V.	1. Configure all used GPIOs to toggle HIGH/LOW under firmware control. 2. Verification passes if for all used GPIOs $VOH \geq 2.4$ V and $VOL \leq 0.4$ V

2.3 Tolerance Analysis

To properly ensure that the device can function without any issues, the power subsystem requires meeting certain tolerances specified by the needs of all the other subsystems, specifically here the processing and sensory subsystems are of most concern since they are active the most during operation.

Component	Max Voltage	Max Current
Microcontroller (ESP32-S3-WROOM-1)	3.3 V	~300 mA
Stretch Sensor	3.3 V	< 1 mA
IMU Sensor	3.6 V	4.5 mA
Calibration Button	24 V	50 mA
Voltage Translator	7 V	128 mA
Translation Transreceiver	4.6 V	100 mA

Table 4: Peak Requirements for device

Component	Max Voltage	Max Current
External Power Connector	48 V	5 A

USB4085-GF-A		
3.3 V Regulator (TLV75533PDBVR)	5.5 V	500 mA
1.8 V Regulator (TLV7551BPDBV)	5.5 V	500 mA
Charging IC (MCP73812T-420I/OT)	4.2 V	500 mA
Battery (LIPO801735 400mAh 3.7V)	4.2 V	600 mA (discharge) 400 mA (charging)

Table 5: Maximum ratings for Power subsystem

Component	Max Voltage	Max Current
Haptic Motor Driver (DRV2605L)	5.5 V	3.5 mA
Piezo-electronic Buzzer (PS1240P02BT)	30 V	-
Red 645 nm LED Indication (151051RS11000)	(Forward Voltage) 2.5 V (Reverse Voltage) 5 V	100 mA
DC Vibration Motor (B1034.FL45-00-015)	3.6 V	60 mA

Table 6: Maximum ratings for the Feedback subsystem

The main analysis to be done is that of the battery's discharging current and the peak requirements for the device; the charging IC safely falls within the battery's peak voltage rating and charging current. The current of the charging IC is programmable, so there is no concern there. Summing the potential max currents of the requirements from the device, we get 582.5 mA, which barely falls under the 600 mA discharge of the battery. Since this peak is covered

within typical operating conditions, it is safe to assume that the device will function without issues. Since the microcontroller's peak current is under the assumption of the wireless functionality being used(which is not for our device), the current use drops considerably into the well safe ranges ~60-100 mA. At the maximum of this current draw, the consumption comes down to 382.5 mA, which is covered by the regulator output currents alone. Under normal operating conditions, our device

3 Costs and Schedule

3.1 Costs

Item	Link	Price	Quantity
PCA9306 (VSSOP)	Link to PCA9306	0.68	2
74AVC4T245D	Link to 74AVC4T245D	0.73	2
ICM 20948 Dev board (Optional used for testing)	Link to ICM 20948	21	1
ICM 20948 Chip	Link to ICM 20948 Chip	6	2
USB4085-GF-A	Link to USB4085-GF-A	0.88	1
TLV75533PDBVR	Link to TLV75533PDBVR	0.36	1
MCP73812T-420I/OT	Link to MCP73812T-420I/OT	0.73	1
LIPO801735 400mAh3.7V	Link to LIPO801735	\$6.95	1
Stretch Sensor 14"	Link to Stretch Sensor 14"	\$20.95	1
DRV2605L -	Link to Haptic Motor Driver	\$11.95	1

Haptic Motor Driver			
DC Vibration Motor	Link to DC Vibration Motor	\$2.95	1
Piezoelectronic Buzzer	Link to Piezoelectronic Buzzer	\$0.78	1
Red 645 nm LED Indication	Link to Red LED Indication	\$0.15	1

Table 8: Purchase List of All Parts Needed for Clip-On Posture Checker

The total cost of the parts shown above is \$81.52, including the optional ICM 20948 Development Board for testing.

According to ZipRecruiter, the average annual salary for someone with a Bachelor's in Computer Engineering is \$117, 751 [11] while the average salary for someone with a Bachelor's in Electrical Engineering is \$107, 650 [12]. These become \$57 and \$52 per hour. Seeing as our project has a team of four people, 3 of whom are in Computer Engineering and 1 of whom is in Electrical Engineering, the breakdown is as follows:

Assuming our team works full-time on this project for the duration of the semester (16 weeks):

$$\$52/\text{hour} * 40 \text{ hours/week} * 16 \text{ weeks} = \$33,280$$

$$3 \text{ people} * (\$57/\text{hour} * 40 \text{ hours/week} * 16 \text{ weeks}) = \$109,440$$

$$\$33,280 + \$109,440 = \$142,720$$

The total payout for our team over the course of creating this device would be roughly \$142,720.

Including the cost of the parts needed to implement our device, the cost would be \$142,801.59.

2.3 Schedule

Week	Task	Group Members
9/22	Divide Tasks	All
	Sensor Subsystem Research	Ashit Anandkumar
	Processing Subsystem Research	Yue Cao
	Feedback Subsystem Research	Destiny Jefferson
	Power Subsystem Research	Edward Ruan
9/29	Component Data Sheet collection	All
	Sensor Subsystem Schematic	Ashit Anandkumar
	Processing Subsystem Schematic	Yue Cao
	Feedback Subsystem Schematic	Destiny Jefferson
	Power Subsystem Schematic	Edward Ruan
10/6	Feedback and Stretch Sensor Breadboard	All
	Sensor Subsystem Schematic	Ashit Anandkumar
	Processing Subsystem Schematic	Yue Cao
	Feedback Subsystem Schematic	Destiny Jefferson
	Power Subsystem Schematic	Edward Ruan
	Work on Design Document	All
10/13	Order Components Required	All
	Assemble PCB	All
	Start PCB testing	All
10/20	Start Programming Microcontroller	Yue Cao
	Continue PCB testing	All
	Revise PCB layout and make necessary edits	All
10/27	Sensor Breadboard	Ashit Anandkumar
	Power Breadboard	Edward Ruan
	Assemble PCB and Test PCB	All
11/3	Continue Testing PCB	All
	Start designing enclosure	All
	Make necessary edits to PCB layout	All
11/10	Final PCB Layout submission if Necessary	All
	Final assembly of all components	All
	Mock Demo Preparation	All
11/17	Mock Demo	All

	Start working on final report	All
	Finalize Presentation	All
12/1	Final Demo	All
	Complete final report	All
	Mock presentation	All
12/8	Final presentation	All
	Submit Final papers	All

Table 9: Budget clip on posture check timeline

4 Ethics and Safety

Ethical Concerns

- E-waste from batteries and other electrical components is a concern.
- The product should be designed to work with diverse body types and clothing styles.

Safety Concerns

- If the device breaks, sharp edges or exposed wires could cause injury.
- The device should be durable enough to withstand drops and moderate stress.
- Battery malfunction could lead to a fire hazard. Fail-safes such as short-circuit protection are needed to mitigate risk of fires.

5 Citations

Bibliography

- [1] E. Y. Duffy et al., “Opportunities to improve cardiovascular health in the new American workplace,” American journal of preventive cardiology, <https://pmc.ncbi.nlm.nih.gov/articles/PMC8315405/#:~:text=Moreover%2C%20more%20than%2080%25%20of,exposure%20to%20physically%20inactive%20behaviors>. (accessed Oct. 13, 2025).
- [2] “Amazon.com: ComfyBrace posture corrector-back brace for men and women- fully adjustable straightener for mid, upper spine support- neck, shoulder, clavicle and back pain relief-breathable : ComfyBrace: Health & Household,” Amazon, <https://www.amazon.com/Corrector-Back-Adjustable-Straightener-Support-Relief-Breathable/dp/B07ZQPKTVV> (accessed Oct. 14, 2025).
- [3] “Desktop app for Live Posture Monitoring and correction,” SitApp, <https://sitapp.app/> (accessed Oct. 13, 2025).
- [4] “Vibration Motor,” SparkFun Electronics, <https://www.sparkfun.com/vibration-motor.html> (accessed Oct. 13, 2025).
- [5] Piezoelectronic buzzers, https://product.tdk.com/en/system/files/dam/doc/product/sw_piezo/sw_piezo/piezo-buzzer/catalog/piezoelectronic_buzzer_ps_en.pdf (accessed Oct. 14, 2025).
- [6] Grainger Engineering Office of Marketing and Communications, “Electrical engineering,” Illinois,

<https://grainger.illinois.edu/academics/undergraduate/majors-and-minors/electrical-engineering> (accessed Oct. 13, 2025).

[7] PCA9554 remote 8-bit I2C and SMBus I/O expander with interrupt output and, https://www.ti.com/lit/ds/symlink/pca9554.pdf?DCM=yes&ds_k=PCA9554&gad_campaignid=14388345080&gad_source=1&ref_url=https://www.ti.com/product/PCA9554?utm_source%3Dgoogle&ts=1759075619617 (accessed Oct. 14, 2025).

[8] TLV755P 500ma, low-IQ, small-size, low-dropout regulator 1 features •, <https://www.ti.com/lit/ds/symlink/tlv755p.pdf> (accessed Oct. 14, 2025).

[9] SN74AVC4T245 dual-bit bus transceiver with configurable voltage ..., <https://www.ti.com/lit/gpn/SN74AVC4T245> (accessed Oct. 14, 2025).

[10] “SN74AVC4T245,” SN74AVC4T245 data sheet, product information and support | TI.com, https://www.ti.com/product/SN74AVC4T245?utm_source=google&utm_medium=cpc&utm_campaign=asc-int-null-44700045336317926_prodfolderdynamic-cpc-pf-google-ww_en_int&utm_content=prodfolddynamic&ds_k=DYNAMIC%2BSEARCH%2BADSDCM=yes&gclid=CjwKCAjw_-3GBhAYEiwAjh9fUAVqBKeP4EYsbnFJySX4AkMNKEjJqN_tv_1gKifxhKHW1z2ExrODLhoCdkgQAvD_BwE (accessed Oct. 13, 2025).

[11] “Salary: Computer Engineering in Illinois (September, 2025),” ZipRecruiter, <https://www.ziprecruiter.com/Salaries/Computer-Engineering-Salary--in-Illinois> (accessed Oct. 14, 2025).

[12] “Salary: Bachelor Electrical Engineering in Illinois (Mar 25),” ZipRecruiter, <https://www.ziprecruiter.com/Salaries/Bachelor-Electrical-Engineering-Salary--in-Illinois> (accessed Oct. 14, 2025).