Anti Back Slouch Reminder

Design Document

ECE 198

Jason Wu, Kelvin Feng

31 October 2023

Table of Contents

Project Overview

As our world becomes more digitalized, it is increasingly necessary for society to prioritize our physical well-being, especially with younger generations. This includes our posture during our activities, like studying posture, walking posture, and more. The Anti Back Slouch Reminder focuses on the user's back posture while seated. Attached to the user will be a flex sensor and an accelerometer, which work to display a slouch angle on an LCD. Under circumstances where the user’s posture is not optimal, the Anti Back Slouch Reminder will buzz and flash LEDs until the user corrects their posture.

Customer Definition

Our project is dedicated to helping people of younger generations improve their back posture. With examinations and busy schedules, many university students neglect the benefits of physical well-being. The environment at university forces many students to sit all day, becoming a health concern for bad back posture if students sit in their rooms for extended periods [1]. Therefore, the Anti Back Slouch Reminder is dedicated to fostering better back posture habits for the 42,000 students studying at the University of Waterloo [2].

Since our project is geared towards promoting good habits for younger generations, the project will be designed for students aged 18 to 23. Recognizing that university students may not have much disposable income, our project is designed to be as affordable as possible for all income classes, and it is in the project’s best interest to keep the product affordable for even low-income classes. For example, 55% of first-year students at the University of Waterloo were classified as low-income [3]. With many students needing to prioritize other financial needs, our product is dedicated to catering to their finances.

Competitive Landscape

Our challenge: make an accurate slouch detection device that encourages users to fix their posture that can be used widely by all students aged 18 to 23 studying at the University of Waterloo while maintaining affordability.

An example of a similar commercially available device

Upright GO2 [4]:

This device uses two movement sensors and is worn as a necklace or is stuck on the back with an adhesive. To prevent slouching, once it is detected, it will begin vibrating to remind the user to sit up straight. It is paired with an app that tracks the slouch of the user over time so that users can see how well they are doing. UWaterloo students would likely prefer wearing these as they do not stick out in an obvious way.

Shortcomings: This device is costly (around $80 USD), and UWaterloo students would likely not want to spend this much on an anti-slouching device. In addition, since it is also placed only on the upper torso, it might not be able to detect bad posture in other parts of the back properly.

Examples of Similar DIY Gadgets

There are plenty of examples of tutorials on the internet for creating slouch detection devices to improve back posture.

Carter Nelson's Accelerometer-Based Slouch Sensor [5]

Carter Nelson describes one such example on Adafruit's website. His design uses an accelerometer built on a device called Circuit Playground. The gadget is wearable and portable and can be put in a shirt pocket or attached elsewhere just under the shoulders. It uses the accelerometer to break down the gravity vector to obtain the angle with the vertical. Slouch is defined here by surpassing a certain angle with the vertical. If the user has slouched for enough time, then it will play a sound to remind the user to correct their posture.

Shortcomings: The device does not fully solve the problem of correcting back posture. For example, the user can hunch their back in a way that is considered bad posture but still maintain an angle with the vertical that will not trigger the alarm. By only measuring this angle, it completely ignores the complex way the human back can curve. In addition, we aim to provide this to all UWaterloo students, and some students may be deaf, so the sound alarm becomes useless. Also, most customers would not want to build the device themselves.

Lara Grant's Flex Sensor-Based Slouch Sensor [6]

Grant's "Slouch-Alert" sensor described on Instructables uses a handmade flex sensor. As users slouch and curve their backs, the flex sensor will change in resistance. Once the values retrieved from the flex sensor surpass an experimentally determined threshold, a motor will vibrate to alert the user to correct their posture. This type of warning system would be able to work for all students, unlike the previous example.

Shortcomings: Given that we require the device to be accurate, using a handmade sensor out of cheaper materials will likely result in errors. Also, this device doesn't entirely solve the slouching problem because it disregards the way people may lean. For example, you can lean in a way with not too much upper back curving, and the flex sensor will not be enough to detect this. Again, most customers don't want to build this themselves.

Requirement Specifications

Functional Requirements

Posture Slouch Detection

The purpose of our product's posture slouch detection function is to detect the angle at which the user is making with respect to other body parts to determine if the user is slouching. To achieve this, we will:

- Calculate the angle between the vertical and the tangent line to the upper torso at the location of the accelerometer
- Detect slouching when the upper torso slumps by 15 degrees or greater. Anything greater than 15 degrees adds stress to the back [7].

Additionally, we will also use a flex sensor to check neck bending. To achieve this, we will:

- Determine the angle by gathering empirical data from the resistance measured on the flex sensor (this value is bound between 10 kΩ and 20 kΩ) [8]
- Detect poor neck posture when the angle reaches 45 degrees or greater, as anything more is deemed unhealthy [9].

Alarm System

The purpose of an alarm system is to catch the user's attention if their posture has reached a level deemed unhealthy. This will be done by:

- Using an array of 1 red 5mm LED light and flashing it on and off in 500 ms intervals
- Using a buzzer to buzz a single alarm at a target decibel of 60dBs produces continuous buzzing noises until the user corrects their posture. This decibel range was chosen to produce noticeable sound while maintaining a non-damaging hearing level [10]

Software Analytics

The purpose of implementing software analytics is to inform the user on how well they have been sitting throughout one use. This analysis will be performed on the microcontroller itself. It will calculate and record:

- Their mean angle of the slouch to an accuracy of 8 degrees within the true mean
- The standard deviation for the angle of the slouch to an accuracy of 8 degrees within the true standard deviation

These true values can be compared through a mathematical analysis of specific test cases.

Technical Requirements

Microcontroller (specifications based on [11])

- Can supply and maintain a voltage of 3.6V
- 1 Mini-USB B to USB 2.0
- Minimum 1 Analog to Digital Converter for the flex sensor
- Minimum 64Kb of RAM
- Minimum 1 I2C sensor for the accelerometer
- Operating frequency of 84MHz

Accelerometer (specifications based on [12])

- 2 μA of current
- Data rate of 1Hz to 5KHz
- I2C or SPI

Flex Sensor (specifications based on [8])

- Needs a pull-up resistor for analog inputs
- Needs a 0.1μF capacitor for digital inputs

LED (specifications based on [13])

- Red coloured
- A maximum current of 20 mA
- Voltage: 1.8-2.3V

Alarm Buzzer (specifications based on [14])

- Voltage range: 3-24 V (larger is preferable for more volume) but rated at 12V
- Required temperatures between -20 and 45 ℃

Software

- Use only numerical data types that require no more than 64 bits for storing calculated values. For example, double-precision floating point numbers may be used.

Safety Requirements

To ensure safety, the voltage being worked with at any point will not exceed 5V to comply with the voltage limit of our microcontroller and sensors. [8] Additionally, the current will not exceed 20mA to ensure LED compatibility [11]. These will be verified using digital multimeters provided by the WEEF lab at the University of Waterloo.

Additionally, the device's buzzer will be limited to no more than 70 dB with a margin of 10 dB. This ensures that the student is not exposed to over 70dB of sound over a prolonged period, as this exposure can start to damage hearing [10]. To measure this requirement, we will use an app called "Decibel Meter Sound Detector" found on the App Store.

Finally, there will be no human or animal test subjects used during the testing and development stages of the device.

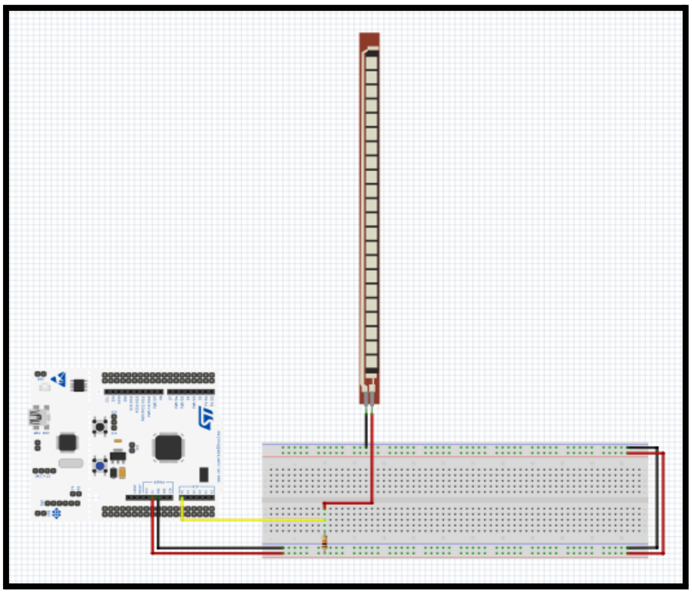
Design

Hardware Description

The Anti Back Slouch Reminder will feature the STM32F401RE microcontroller, a flex sensor, an accelerometer, an LCD, a buzzer, and an LED, all of which will interact through a breadboard and can be powered via a mini USB-B to USB 2.0 connector onto a laptop/computer or any 5V DC power supply. The microcontroller reads voltage input from the flex sensor through analog inputs. The accelerometer communicates with the microcontroller through SPI wiring. The LCD communicates with the microcontroller through I2C wiring. During operation, the microcontroller runs software to actively monitor and collect data regarding the user's slouch angle. The LCD will receive the appropriate input to display this information on the screen. Once a poor level of slouch is detected, the microcontroller will sound the buzzer and light up the LED until the slouch angle returns to an appropriate level.

Microcontroller Wiring Step-by-Step

Note: Skip to Checkpoint 3 for the final circuit diagram reference

<table>
<tr>
<td>Connect the following Board Pins to the Breadboard:
Board 5V to Breadboard Power Rail (+)
Board GND to Breadboard Power Rail (-)
Breadboard Power Rail (+ lower) to Breadboard Power Rail (+ upper)
Breadboard Power Rail (- lower) to Breadboard Power Rail (- upper)</td>
<td>Connect the following Flex Sensor Pins:
Flex Sensor Negative to Breadboard Power Rail (-)
Flex Sensor Positive to Breadboard 10E
Breadboard 10A to Breadboard Power Rail (+) using 10k Ωresistor
Breadboard 10C to Board A0</td>
</tr>
<tr>
<td colspan="2"><u>Checkpoint 1: The device should look similar to the image:</u>
</td>
</tr>
</table>

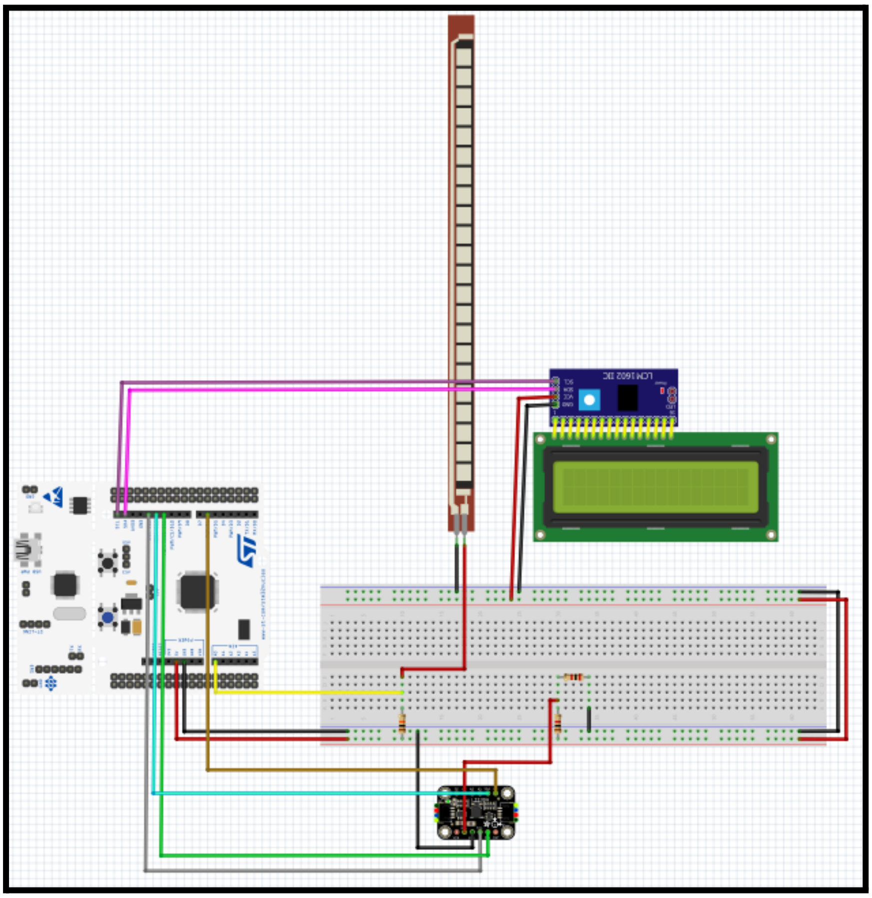
Connect the following Accelerometer Pins:

- Accelerometer 3V3 to Breadboard 30B
- Accelerometer GND to Breadboard Power Rail (-)
- Accelerometer SCL to Board SCK
- Accelerometer SDA to Board MOSI
- Accelerometer SDO to Board MISO
- Accelerometer CS to Board PWM/D6
- Breadboard 30A to Breadboard Power Rail (+) with 1k Ω resistor
- Breadboard 30E to Breadboard 34E with 2k Ω resistor
- Breadboard 34A to Breadboard Power Rail (-)

Connect the following LCD Pins:

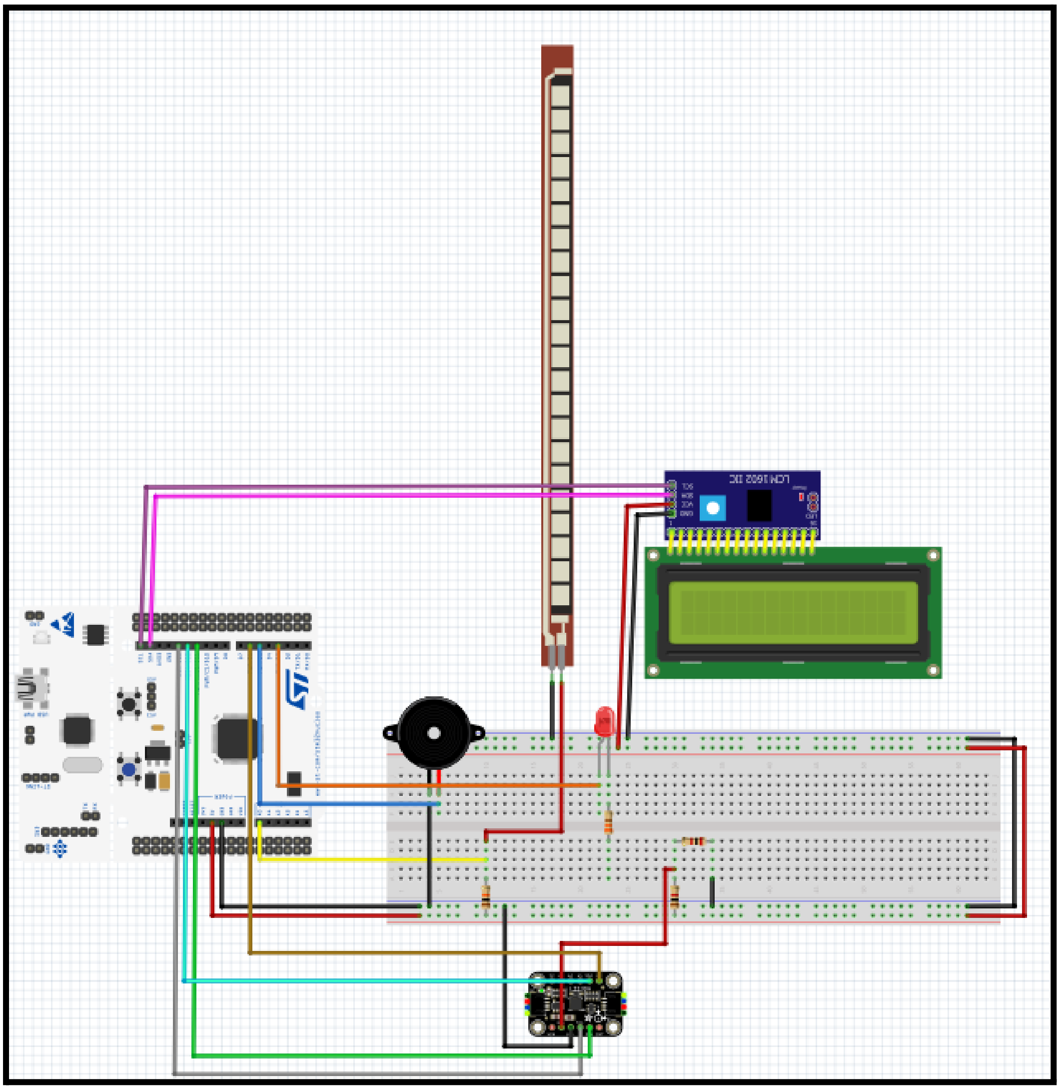
- LCD GND to Breadboard Power Rail (-)
- LCD VCC to Breadboard Power Rail (+)
- LCD SDA to Board SDA
- LCD SCL to Board SCL

<u>Checkpoint 2: The device should look similar to the image below:</u>



Connect the following Buzzer Pins:	*Connect the following LED pins:*
• Buzzer (-) to Breadboard 4H • Buzzer (+) to Breadboard 5H • Breadboard 4G to Breadboard Power Rail (-) • Breadboard 5G to Board PWM/D5	• LED anode to Breadboard 22J • LED cathode to Breadboard 23J • Breadboard 22I to Board PWM/D3 • Breadboard 23G to Breadboard 23E using 330Ω resistor • Breadboard 23C to Breadboard Power Rail (-)

<u>Checkpoint 3: The final circuit design of the Anti Back Slouch Reminder:</u>



Software Description

The STM32F401RE will be programmed in C++ and will use Adafruit libraries, LCD libraries from the internet for display, sections of the C++ standard library (for mathematical purposes), and libraries provided by STMicroelectronics. The software will retrieve data from the flex sensor and accelerometer and use STMicro libraries to calculate the mean every 0.1 seconds. Software Pseudocode & Details:

Please take into account that the pseudocode aims to illustrate each component, and in practice, further changes must be made to make everything compatible.

Also, note that this pseudocode uses Let name := description for definitions and a single = for assignment.

We will not include all the full details for implementing it but rather provide a sense of how the logic behind it works. Double-precision floating point numbers will be used for all numerical values.

We use `*` for multiplication and `/` for division.

These are the definitions of the less clear values and variables described in the pseudocodes

```
Constants:
Let collection_rate := how often data should be collected from
the sensors for calculating mean and std deviation
--------------------
Let prev_time := last checked time (in ms)
Let curr_time := current time (in ms)
Let sum_sq_theta := sum of the squares of the angle values (in
radians).
Let sum_theta := sum of the angle values in radians.
Let n_theta := number of angles collected
```

Data Analytics

Our methods for calculating the mean and standard deviation take into account the methods mentioned in [15]

Straightforward calculation method adapted from Jonathan Leffer [15]:

For calculating standard deviation, we will use the following formula:

Let μ be the mean, n be the number of data values, x_i be the ith data value, σ be the standard deviation

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} x_i^2 - n\mu^2}{n}} = \sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n} - \mu^2}$$

This method continuously adds to local variables holding the sum of all values and the sum of the squares of all values.

```
while program is running:
      curr_time = get_current_system_time()
      Collect all relevant sensor data (curr_angle, curr_flex)
      If curr_time - prev_time >= collection_rate:
            Add curr_angle to sum_theta
            Add (curr_angle x curr_angle) to sum_sq_theta
            Avg = ((sum_theta)*(num_theta/(num_theta +1)) + ((curr_angle)/ (num_theta + 1))

            Comment: doing the operations in this order will
            likely help counter the precision issues (losing values in
            decimal places) with floats. This uses the formula given by
            Abdullah Al-Ageel [16]

            Increment num_theta by 1
            Std_dev = sqrt((sum_sq_theta/num_theta) - (avg*avg))
```

These values calculated can later be outputted to the LCD.

Welford's Algorithm (adapted from [17])

Welford's algorithm can be used for updating running standard deviations and averages when new values are added, and the current mean, standard deviation, and number of values are known and precalculated. Welford's algorithm presents the formulas for the mean and standard deviation in a recurrence relation that is numerically stable and compensates for the imprecision of floating point numbers.

It uses two formulas:

1. Mean

$$\mu_{n+1} = \mu_n + \frac{1}{n+1}(x_{n+1} - \mu_n)$$

2. Standard Deviation

$$\sigma_{n+1} = \sqrt{\sigma_n^{\ 2} + \frac{(x_{n+1}-\mu_n)(x_{n+1}-\mu_{n+1}) - \sigma_n^{\ 2}}{n+1}}$$

In this case, the pseudocode will look like the following:

```
Let prev_avg := a copy of the (soon to be previous) average

while program is running:
	curr_time = get_current_system_time()
	Collect all relevant sensor data (curr_angle, curr_flex)
	If curr_time - prev_time >= collection_rate:
		prev_avg = avg
		Avg = avg + (curr_angle-avg)/(num_theta+1)

		Std_dev = sqrt(std_dev*std_dev + (((curr_angle -
		prev_avg)*(curr_angle - avg) -
		(std_dev*std_dev))/(num_theta+1))

		Increment num_theta by 1
```

Sensor data processing

Fetching the sensor values can quickly be done with Adafruit's libraries.

Decision-making based on sensor data pseudocode:

Essentially, we use simple if statements (conditionals) to check for slouching and then sound the alarm and flash the lights if it is detected

```
if(curr_flex >= flex_threshold OR angle >= angle_threshold):
	Sound the alarm
	if(time since last flash is over 0.5 s or haven't started
```

```
    flashing):
        Flash the led again
```

Interpreting accelerometer values to produce the angle:

We will put this all in a function.

```
Function get_angle(z_comp_acc):
    Return arcsin(-z_comp_acc/9.81)
```

Note: 9.81 is the value we have chosen for gravitational acceleration.

Science and Mathematical Principles

Certain scientific and mathematical principles will be implemented to ensure this project's features are feasible.

1. **Ohm's Law**

A flex sensor can measure the angle created to ensure accurate student posture data. Flex sensors work by changing resistance based on their level of flex, which directly affects the sensor's resistance. We can calculate the current using Ohm's Law, V=IR [18]. By measuring the current, we can determine the degree to which the student is bending their back, thus determining the severity of their posture. In this scenario, adjusting the voltage is the key design parameter, as it acts as a measure of the user's slouch angle. Ohm's law is also crucial for determining the resistance required to ensure that our parts can operate. For example, the LEDs that we plan on using require that the maximum current must not exceed 20mA. To achieve this, we can use Ohm's Law to calculate the minimum resistance required.

We know that the maximum voltage that the microcontroller can output is 3.3V. The maximum current cannot exceed 20mA.

$V = IR$

$R = V / I$

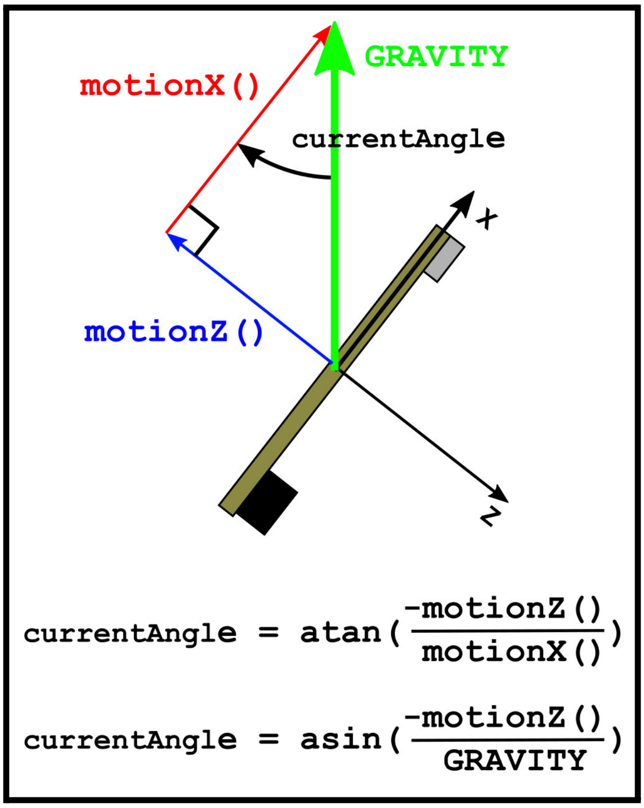
$R = 3.3V / 0.02A$

$R = 165\Omega$

Therefore a resistor of at least 165Ω is needed to protect the LEDs.

2. **Vector Components and Trigonometry**

The following method is based on Adafruit's methods of using an accelerometer to find the angle with the vertical [5]. While using the accelerometer, breaking up the force of the gravity vector into its horizontal and vertical vector components is necessary. From there, the primary trigonometric functions and their ratios for a right-angle triangle are required to determine the angle. A forces diagram from Adafruit is provided below to complement this explanation:



The accelerometer can provide us with the x, y, and z acceleration components. Let us denote these as a_x, a_y, a_z.

Let the angle of interest (between the vertical) be denoted θ

Since the user is sitting, we will only be measuring gravity.

We form a triangle using the gravity vector and a_x and a_z (which are components of it)

Let g be the gravitational acceleration constant (9.81 m/s^2)

Using trigonometry, we have

$sin\theta \; = \; \frac{|a_z|}{g}$ so

$\theta \; = \; sin^{-1}(\frac{|a_z|}{g})$

If we want to display this value back to the user since the standard library for C++ uses radians, we will need to change it to the more user-friendly units of degrees so we display the value given by $\theta_{rad}(\frac{180°}{\pi})$

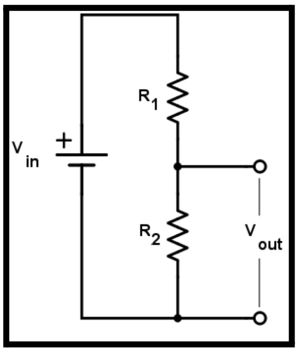
Example calculation:

Suppose a_z = -5.01, then $\theta_{rad} = sin^{-1}(\frac{|-5.01|}{9.81}) = 0.536$

In degrees, the angle is $\theta_{deg} = \theta_{rad}(\frac{180°}{\pi}) = 30.71°$

3. Voltage Divider Circuit

A voltage divider may be necessary to manage the voltage between the microcontroller and our external components for additional safety. Additionally, to complement our resistance readings from the flex sensor, a resistive sensor can be connected to a voltage divider alongside a resistor to interface with the microcontroller [19]. A voltage divider schematic is provided below:



In this scenario, R2 will represent the resistance from the flex sensor. To properly calculate the Vout, we can use the equation Vout =$\frac{R2}{R1+R2}$ (Vin).

Example Calculation:

Suppose R2 = 12kΩ, R1 = 10kΩ, and Vin = 3.3V.

Therefore Vout = $\frac{12k\Omega}{10k\Omega+12k\Omega}$(3.3V) = 1.8V

Vout can be interpreted by the microcontroller, and after calibration, an angle can further be interpreted within the microcontroller.

4. Statistical Analysis: Standard Deviation, Mean of a Data Set, and Linear Regression

$$\sigma = \sqrt{\frac{\Sigma(x_i-\mu)^2}{N}} \qquad \bar{x} = \frac{1}{N}\Sigma x_i$$

By measuring their mean slouch angle and the standard deviation for the slouch angle, users can keep track of how well their posture was over the day. Since standard deviation describes the spread of the data with respect to the mean of the dataset [20] and since the outliers can often throw off the mean, the standard deviation will be able to tell the user how closely they actually stuck to their mean. This means that with a "good" mean angle, if the standard deviation value is low, they have consistently had good posture. In addition, we will log their mean slouch angle every day, then perform linear regression on the dataset with time on the x-axis so that they can visually view how they are improving/worsening over time.

Manufacturing Costs

List of materials and technologies required. No special equipment is required to build this device.

Material	Manufacturer and Location	Distributor and Location	Source
STM32F401RE Microcontroller	STMicroelectronics, Headquartered in Switzerland, Assembled in China	University of Waterloo - Wstore, Canada	[11]
Long Flex Sensor	Adafruit, Manufactured in the United States	Adafruit, United States	[8]
Accelerometer	Adafruit, Manufactured in the United States	Adafruit, United States	[12]
LCD (with I2C compatibility)	WayinTop, Manufactured in the United States	Amazon, Canada	[21]
Buzzer	Qianxin, Manufactured in China	Amazon, Canada	[14]
Breadboard and Wires	Haobase, Manufactured in China	Amazon, Canada	[22]
Resistors (1x10kΩ, 1x2kΩ, 1x1kΩ, 1x330Ω)	N/A	University of Waterloo - Ridgidware, Canada	N/A
LED	N/A	University of Waterloo - Ridgidware, Canada	N/A

Implementation Costs

Installation Manual

Welcome to the Anti Back Slouch Reminder installation manual. This manual will guide you through the setup process step by step.

1. *Check for loose wires*

Observe if there are any loose or hanging wires and resistors. In particular, ensure that the 1k and 2k ohm resistors are securely in place. This prevents the voltage from exceeding 3.3V, which is necessary for the product to work. An image of the resistors is provided below. If the colour strips don't match, DO NOT power on the device.

1k ohm resistor	2k ohm resistor
	

If the resistors and wires are secure, proceed to step 2.

2. *Attach the accelerometer and flex sensor to the back of a shirt*

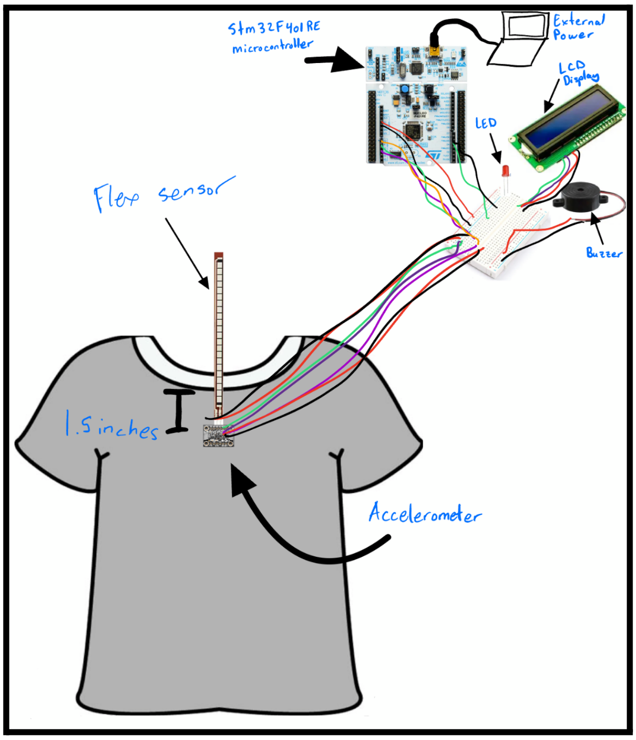
The next step is to attach the accelerometer and flex sensor to a shirt. First, take the flex sensor and line it up in the middle on the back side of the shirt. Apply adhesive to stick 1.5 inches of flex sensor on the shirt. Next, take the accelerometer and apply adhesive directly below the flex sensor. Lastly, put on the shirt and apply adhesive to stick the top of the flex sensor to your neck.

3. *Power the anti-back slouch reminder.*

After installing the accelerometer and flex sensor, we can power the device.

Important: before connecting to power, ensure you are already sitting up straight. After ensuring proper initial posture, connect the device to a computer using the provided mini USB-B to USB 2.0 cable or connect the device to a 5V DC power supply of your choice.

And you're all set up! If done correctly, the setup should look similar to the image shown below:



Note: This diagram is NOT accurate in size, nor is any of the electrical wiring. The intended purpose of this visual is to demonstrate how the project should be set up in terms of the location of interacting parts.

User Guide

Welcome to the Anti Back Slouch Reminder user guide. This guide will outline all the features of our product and how to use it to its full effect.

1. *Core Function*

From the moment the device is powered on, the device will start reading your posture. This is done by looking at the angle of the slouch created on your back. The result of this angle will be visible to you through the LCD screen (in degrees).

2. *Bad Posture Detection*

If the buzzer starts to ring and the LED starts to flash on, the device has detected poor posture. Poor posture is determined when your angle exceeds 15 degrees. To turn off the ringing and flashing, simply correct your posture to be below 15 degrees.

3. *Average Posture Display*

While the device is in use, the average degree of slouch will also display on the LCD screen. This average is taken from the moment the device was powered on. This is a useful indicator of long-term improvements in posture.

Energy Analysis

According to initial project requirements, the device may not consume, transfer, discharge, or expend more than 30W of power at any time.

The baseline power level is provided through a 5V DC Power Adapter. This will be done using USB cables, provided through the purchase of the STM32F401RE microcontroller.

Evidence: according to [23], these USB cables will not exceed a voltage of 5V. In addition, 5V is appropriate because according to [11], the STM32 microcontroller takes a 5V input.

Power Consumption

To calculate the power consumption, we can use the equation P = IV, where P indicates power, I indicates current, and V indicates voltage. Combining this with Ohm's law, we can calculate the total power of each component and then take the sum to ensure that it meets standards.

<u>Power consumption of STM32F401RE Microcontroller:</u>

Max Voltage: 12V [11]

Max Current: 250mA [11]

P = IV = 12(.25) = 3W

<u>Power consumption of Accelerometer:</u>

Max Voltage: 5V

Max Current: 4μA [12]

P = IV = 5(0.000004) = 0.00002W

<u>Power consumption of LCD:</u>

Max Voltage: 5V

Max Current: 160mA [21]

P = IV = 5(.160) = 0.8W

<u>Power consumption of Buzzer:</u>

Max Voltage: 5V

Max Current: 10mA [14]

P = IV = 5(0.01) = 0.05W

<u>Power consumption of LED:</u>

Max Voltage: 5V

Max Current: 20mA [13]

P = IV = 5(0.02) = 0.1W

Sum of Power = 3W + 0.00002W + 0.8W + 0.05W + 0.1W = 3.95002W

Since our sum does not exceed 30W, our project meets this standard.

Stored Energy

To consider stored energy, we can look at the capacitors on the microcontroller and the LCD, as they are the only components capable of storing electrical energy.

Energy storage of STM32F401RE Microcontroller:

Taking the largest capacitor on the board, a 2.2μf capacitor [11], by using the energy in a capacitor equation, we can obtain the maximum stored energy in the capacitor.

E = ½(Q)(V)^2, where Q = the capacitance and V = the voltage around the capacitor

E = ½(0.0000022f)(5V)^2 = 0.0000275J = 0.0275mJ

On the board, there are ten capacitors [11], each of varying capacitance but not exceeding the maximum capacitance of 2.2μf. Assuming that all capacitors can hold the same amount of energy (although they don't, we assume this for the sake of ensuring that the system does not exceed the maximum), we can multiply our energy by 10 to receive an overestimated maximum energy storage. Energy Storage = 10E = 0.275mJ.

Additionally, there are no chemical forms of energy. There is mechanical motion involving the flex sensor, however, the mechanical motion is not elastic, meaning it does not store elastic potential energy.

Energy storage of LCD:

On the LCD, there are two capacitors, each with a capacitance of 400pf [21]. By using the same equation above, we can obtain the maximum stored energy.

2E = QV^2 = (0.0000000004)(5)^2 = 0.00000001J = 0.00001mJ

Energy Storage = 2E = 0.00001mJ

Since there are no additional components that store energy, the maximum energy that could be stored is 0.27501mJ, which is less than the maximum of 500mJ, thus satisfying the requirement.

Risk Analysis

Negative consequences of using the product as intended:

- Every person's body is different; the angle for a "normal" posture may vary from person to person, so users may be forced into uncomfortable postures which are unideal for their health.
- The product is not built to be "green," so the manufacturing of the components contributes greenhouse gases, and since it requires electricity to run, this is also not environmentally friendly. In addition, this is not designed to be recyclable, so it will likely end up in the landfill after its lifespan finishes.
- Despite our product only measuring angle, an angle is not the most accurate way of interpreting posture, so users may deceive themselves into assuming a posture that is unhealthy for them yet satisfies the device

Negative consequences of using the product incorrectly:

- If the user does not power on the device already sitting straight (as outlined in the User guide), then the accelerometer will not be properly calibrated and the readings will be inaccurate, thus the device may encourage the user to assume a position bad for their back
- Small components on the device can pose choking hazards to young children if left unattended
- Moving away too quickly from the desk with the device still attached could lead to hazards from flinging the alarm system and microcontroller off the desk.
- If the user puts it on a body part it is not intended to go on, then they risk wire tanglings and higher probabilities that the wires snag onto something and knock things down.
- If the user supplies the devices with a power source greater than 5V, then the device will break and may deliver an electric shock to them.

Negative consequences of misusing the design:

- The product could be intentionally abused to mislead other people into believing that their posture is bad. This could be done if a user intentionally avoids calibrating the device, messing up the readings. This then encourages them to assume postures are bad for their back health to "fix it."

- The long wires in the design could be repurposed as trip wires to hurt people or repurposed as whips.

Possible ways to malfunction:

- There is a failure in reading the sensors accurately
- The product will not be waterproof; the system will malfunction if the circuitry encounters any liquids.
- Since many wires will be used, the device could malfunction if the user accidentally knocks over a wire with their arm. In the worst-case scenario, this could knock the device to the ground, disconnecting all components.

Consequences for each of the failure mechanisms specified:

- With incorrect readings, the user can be misled into false positives/negatives for bad posture
    - For example, if the product tells the user that they have good posture, but they don't, then the user won't correct their posture (health risks)
- The product can cease to function if water manages to short-circuit the product and it may create sparks (which is concerning if paper is nearby)
- The product can cease to function if it gets knocked over in the worst-case scenario. It may also create additional hazards by flinging the parts of the table.

Test Plan

Requirement	Test setup	Environmental parameters	Test inputs	Pass Criteria and quantifiable measurement standard
Detect slouching if the upper torso leans by 15 degrees or more	Using a box, label one side of the flaps with angles from 0 - 90 degrees, centred at the rotation axis of the adjacent flap, with the vertical axis at 0 degrees. Attach	A room where we are allowed to make noise without disturbing others Outlet or power source nearby Between -20 and 45 degrees	The box flap will be rotated through the angles from 0 to 12 degrees (to ensure no false alarms) Then from 12 onwards to 60 degrees (to	The alarm will sound upon reaching an angle of 15 degrees or more (the standard) The test passes if a) No false alarms between 0 to 11 degrees b) Alarm sets off upon reaching ±3

	the sensor onto the rotating flap vertically (so it should be at 0 degrees). Once power is applied, rotate the flap back and forth to simulate a change in slouch. Must bring a protractor.	Celsius (to allow the alarm to be operational), preferably at room temperature	ensure the alarm will be sounded)	degrees from the standard target angle c) Alarm continues to alert from 15 to 60 degrees
Detect if the alarm hits the target of 60dB at room temperature	Download an app that can detect the intensity of sounds in decibels, then allow it to begin recording Mount the system on the box flap and place the alarm system on a table. Orient it vertically before supplying power.	Quiet, relatively private area (to filter out external noises) Room temperature (20±5) ℃ Permission to make noise in the area Nearby power source	Angle to the vertical is tilted more than 15 degrees 3 trials done	The target standard: reaching 60 dB at room temperature in a small room The device passes the test if a) The app detects a sound intensity of (60±5) dB from a realistic distance (buzzer on table, dB measured from distance to ear) in at least 2 of 3 trials b) The sound intensity does not exceed 70 dB
Detect neck slouching of 45 degrees or more	Using the same box described above, attach the sensor below the flap, with half of the flex sensor above the flap. Attach the tip of the flex sensor to the flap. The box setup is the same as described above,	Permission to make noises (from the alarm) Nearby power source Between -20 and 45 degrees Celsius (to allow the alarm to be operational), preferably at room	3 trials of Bend again at 45 Then, Bend at 20, 30, 40, 50, and 60 (5 trials altogether)	Target standard: Alarm sets off upon reaching the corresponding empirically determined flex sensor unit value to the 45-degree neck bend Passes test if: a) It consistently sets off the alarm at 45 degrees for each of the three trials

	with angles prelabelled on the adjacent flap.	temperature		b) The earliest it sets off at is within 5 degrees of the standard target (set off earliest at 40 or 50) c) It continues to sound the alarm at larger angles 60 degrees
5V max voltage	Get a multimeter with probes. Ensure it is undamaged and accurate Set up the system in full on a table but do not need to orient it in any specific way or bend the sensor	No dust nearby [24] Room temperature (20±5) ℃ Nearby power source	5V supplied from the power source to the system. Capacitors could possibly make it exceed this.	Target standard: No component of the system exceeds 5V Passes tests if: a) Nowhere in the system exceeds a voltage of 5V (allowance of 0.1V for measurement errors)
LED: 20 mA max current, 1.8V-2.3V	Same set up as above. Ensure resistors are secured properly. This test should be done first.	Same environmental parameters as above	5V supplied from power source to the system	Target standard: The current in any place going in or out of the LED does not exceed 20mA and voltage is between 1.8V - 2.3V Passes test if: a) Current in and out of LED is less than 20 mA (cannot be higher) b) Voltage fits within the range (no allowance for a bit over or under)
Software analytics accuracy	Perform a mathematical analysis with the given inputs to figure out what the true std dev	A room where we are allowed to make noise without disturbing others	Angles and times they are held for may be determined right beforehand. For example, try	Standard and pass criteria: true std dev and mean values are within 8 degrees of the mathematically calculated ones

	and mean. Create an experimental setup like the one described in the first test in the table. Have someone ready to film the test to recall std dev and mean values in real time. Have a timer handy.	Outlet or power source nearby Between -20 and 45 degrees Celsius (to allow the alarm to be operational), preferably at room temperature	analyzing 10 degrees for 2s 30 degrees for 3s 20 degrees for 1s	

References

[1] R. G. Morse. “Why does my lower back hurt when I sit and how can I relieve the pain?” Healthline. https://www.healthline.com/health/lower-back-pain-when-sitting (accessed September 20, 2023).

[2] “Quick facts | About Waterloo.” University of Waterloo. https://uwaterloo.ca/about/who-we-are/facts (accessed September 20, 2023).

[3] “University of Waterloo PA-8: Affordability and Access.” AASHE. https://reports.aashe.org/institutions/university-of-waterloo-on/report/2021-10-26/PA/diversity-affordability/PA-8/ (accessed October 28, 2023).

[4] J. Powell. “Upright Trainer Review.” Healthline. https://www.healthline.com/health/upright-trainer-review (accessed October 20, 2023).

[5] C. Nelson. “Measuring slouch angle.” Adafruit. https://learn.adafruit.com/circuit-playground-slouch-detector/measuring-slouch-angle (accessed September 25, 2023).

[6] L. Grant. “Slouch Alert.” Instructables. https://www.instructables.com/Slouch-Alert/ (accessed October 20, 2023).

[7] M. Hecht. “The 4 main types of posture.” Healthline. https://www.healthline.com/health/bone-health/the-4-main-types-of-posture (accessed September 26, 2023).

[8] “Long flex sensor.” Adafruit. https://www.adafruit.com/product/182 (accessed September 27, 2023).

[9] “The dangers of tech neck.” Choice Pain & Rehabilitation Center. https://www.choicepain.com/blog/the-dangers-of-tech-neck (accessed September 29, 2023)

[10] “What noises cause hearing loss?” Centers for Disease Control and Prevention. https://www.cdc.gov/nceh/hearing_loss/what_noises_cause_hearing_loss.html (accessed September 28, 2023).

[11] STMicroelectronics, “STM32F401xD STM32F401xE,” STM32F401 datasheet,

January 16, 2014 [Revised January 22, 2015].

[12] "Adafruit LIS3DH triple-axis accelerometer." Adafruit. https://www.adafruit.com/product/2809 (accessed September 28, 2023).

[13] Eagle Smart Tech. "500PCS 5 Millimeter LED Light Emitting Diode Assortment Kit, Low Voltage Diffused Diode for DIY PCB Circuit, Red, Yellow, Green, Blue and White LED Indicator Lights." Amazon. https://www.amazon.ca/Millimeter-Emitting-Assortment-Diffused-Indicator/dp/B07N2GVCYZ/ (accessed September 19, 2023).

[14] Aloway. "Cylewet 5Pcs SFM-27 DC 3-24V Electronic Buzzer Alarm Sounder Continuous Sound Beep (Pack of 5) CLT1083." Amazon. https://www.amazon.ca/Cylewet-Electronic-Sounder-Continuous-Intermittent/dp/B07HDYSY1V (accessed September 19, 2023).

[15] J. Leffer and A. Reynolds. "How to efficiently calculate a running standard deviation." Stack Overflow. https://stackoverflow.com/questions/1174984/how-to-efficiently-calculate-a-running-standard-deviation (accessed October 29, 2023).

[16] A. Al-Ageel et. al. "How to calculate a moving average without keeping the count and data-total." Stack Overflow. https://stackoverflow.com/questions/12636613/how-to-calculate-moving-average-without-keeping-the-count-and-data-total (accessed October 29, 2023).

[17] C. Chen. "Welford's algorithm for updating variance." Changyao Chen. https://changyaochen.github.io/welford/ (accessed October 29, 2023).

[18] "Ohm's law." Britannica. https://www.britannica.com/science/Ohms-law (accessed September 28, 2023).

[19] "Voltage divider calculator." All About Circuits. https://www.allaboutcircuits.com/tools/voltage-divider-calculator/ (accessed September 19, 2023).

[20] "Standard deviation." National Library of Medicine. https://www.nlm.nih.gov/oet/ed

/stats/02-900.html (accessed September 26, 2023).

[21] WayinTop-US. "WayinTop 20x4 2004 LCD Display Module with IIC/I2C/TWI Serial Interface Adapter (Blue/2004)." Amazon. https://www.amazon.ca/WayinTop-Display-Interface-Adapter-Arduino/dp/B07TXGD3WS/ (accessed October 10, 2023).

[22] NBCA. "Haobase Solderless Breadboard Bread Board 830 Tie Points Contacts with 65PCS Male to Male Solderless Breadboard Jumper Cable". Amazon. https://www.amazon.ca/Haobase-Solderless-Breadboard-Points-Contacts/dp/B01FJ7FJKO/ (accessed October 10, 2023).

[23] "USB Charging and Power Delivery." Tripp Lite Eaton. https://tripplite.eaton.com/products/usb-charging (accessed October 29, 2023).

[24] Y. S. Chan. "Why Digital Multimeter Accuracy and Precision Matter." KAiWEETS. https://kaiweets.com/en-ca/blogs/news/why-digital-multimeter-accuracy-and-precision-matter (accessed October 29, 2023).