Back to Healthy Posture

Design Document

Daniel Chen, Fangqi Han, Christian Held
Group 10

ECE 445

April 17 2020

Table of Contents

1. Introduction

1.1. Problem and Solution Overview

1.1.1. Many people do not sit correctly and have bad posture. This can lead to back problems, pains, and potentially cause a medical issue. There should be a way to monitor one's posture and have feedback for their sitting position over the course of a work or study day. There are many components of good posture, such as lack of movement, kyphosed, slouching and imbalanced sitting posture [1] [4].

1.1.2. The system will consist of 8 or more pressure sensors, 1 distance sensor, and an app that the user can download. This apparatus will be draped over the seat where the user sits. It will measure the force applied to the backrest of the chair and to the seat to see if proper and even pressure is applied by the seating posture.

1.2. **Visual Aid**

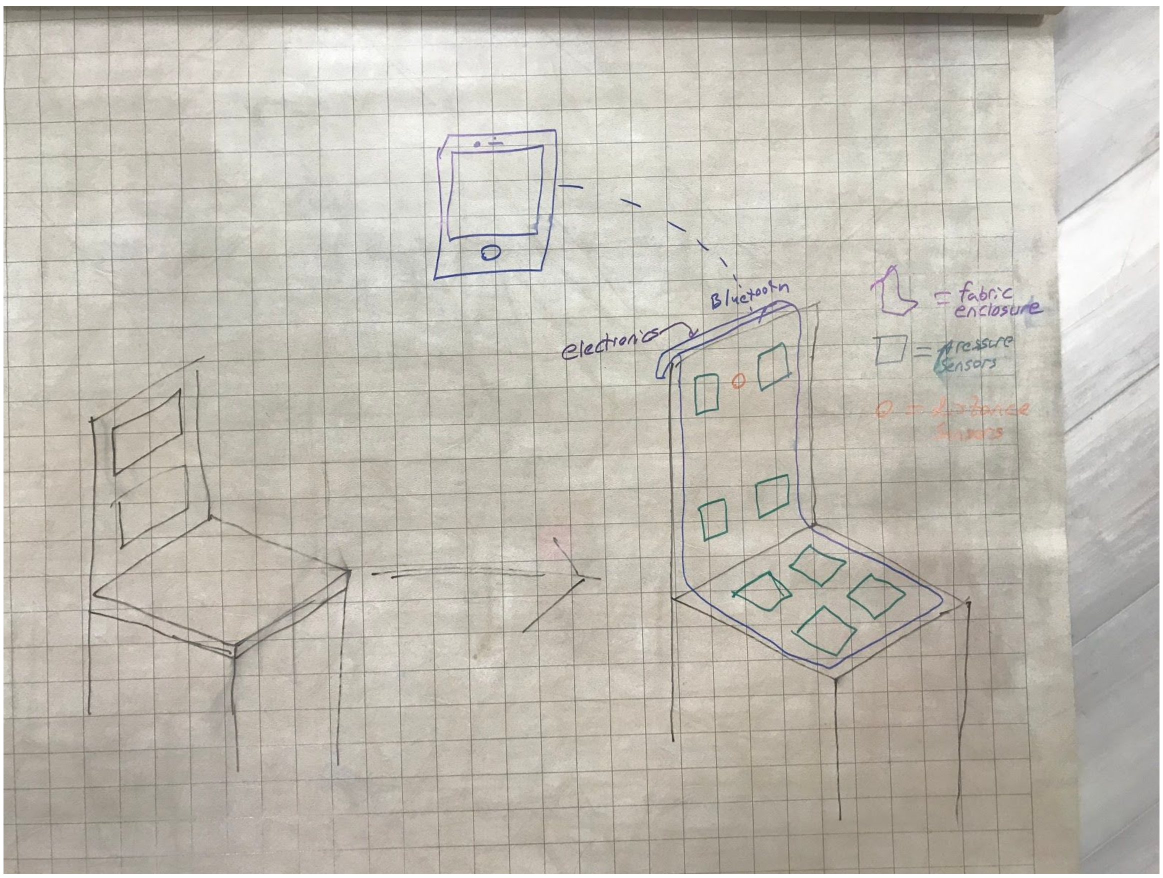

Figure 1. Visual Aid

The product is applied to a standard chair design in order to operate. The user then connects to the device via their android application. They run through the setup and then sit and pay attention to their posture as they use the chair.

1.3. **High-level requirements list**

1.3.1. The user needs to be able to check **pressure** data calculated by the system in real time with an accuracy of 0.5 kilogram.

1.3.2. The user needs to be able to check **distance** data calculated by the system in real time with an accuracy of 1 centimeter.

1.3.3. The user needs to be able to adjust parameters to fit their personal data and check their feedback using a phone application as interface.

2. Design

2.1. **Block Diagram**

Figure 2. Block Diagram

After the battery voltage is regulated the microcontroller reads the data from the eight pressure sensors and one distance sensor and packages the data to be sent off via the bluetooth module. The module then sends the data to the phone application. The phone application takes the raw data and converts it into a display for the user to actively monitor their seating position.

2.2. Schematic

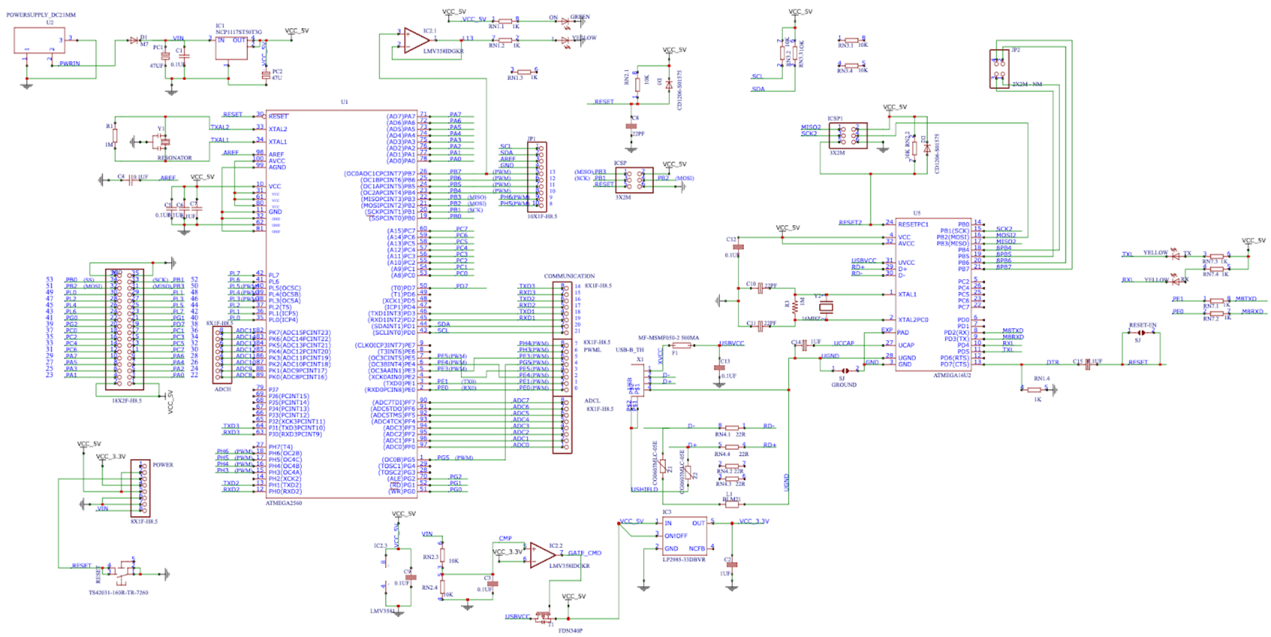

Figure 3. Arduino Schematic

Figure 3 shows the MEGA schematic provided by Arduino [5]. The Arduino MEGA is what our microcontroller design is based off of. A lot of the circuitry can be eliminated as most of the pins are unused and the MEGA has many functions that are not needed for our application. This mainly includes: multiple clock frequencies, all pins providing separate PWM waves, and digital to analog conversions.

2.3. Pressure Sensors

2.3.1. These devices measure what force the user is applying to cushion and send the data to the microcontroller. They are powered from the voltage regulators. They nominally have a very large resistance (manufacturer says “approximately infinite”), but when depressed they change to a lower resistance. Using a voltage divider, one can calculate the resistance and convert that to a pressure amount.

2.4. Battery

2.4.1. This supplies power to the device. The voltage is nominally 7.4 volts from the battery, but it can range from 6-8.8 volts. The decision was made to use a lithium polymer (LiPo) battery due to the longevity of the charge from a LiPo battery. The voltage is a bit too high for the devices and so the power is first passed through the voltage regulator.

2.5. Voltage Regulators

2.5.1. These convert the battery voltage to the correct level for each component. For battery safety reasons the circuitry will shut off the system if there is too low of a voltage (UVLO). The voltage we will run on will be 5 volts for the pressure sensors, microcontroller and distance sensors and they are all sensitive to variations in voltage so the 7.4 volts must be converted. For safety reasons, multiple voltage regulators will be used because the sensors will be in direct contact with the user and we would like to limit the maximum current through. On the other hand, we would like to limit the current to the microcontroller to a lesser degree.

2.6. Microcontroller

2.6.1. Interprets data from pressure sensors and distance sensors and sends the raw data to the phone application via a bluetooth module. The structure of the microcontroller will be an Arduino Mega chip and using their design suggestions.

2.7. Bluetooth Module

2.7.1. The bluetooth module should transmit data between the microcontroller and the phone app. The module we have selected supports up to Bluetooth 4.0 which is sufficient for our needs.

2.8. Phone App

2.8.1. The phone app serves as a user interface. It should allow input and storage of baseline data from the microcontroller for postures the user considers as healthy. It will also evaluate data sent from the microcontroller via bluetooth and warn the user of bad postures in real time. This will determine what is a good posture when the user goes through the process of setting up the application each sitting session. This is to account for individuals who deviate from the “normal” sitting position (due to an amputation, bone defect, etc.).

2.9. Distance sensor

2.9.1. This device is mounted towards where the upper back of the user rests. It measures the distance of the upper back and helps determine if the user is in a compromising position. The device uses light from an IR led and measures the angle of the reflected beam with a PSD (position sensitive device) to determine distance.

2.10. Soft Pad

2.10.1. Simply a swapable pad that the user can remove. The user can adjust the firmness of the device by swapping to a different pad. Additionally this allows the user to remove an element of the device for hygienic reasons.

2.11. Fabric Enclosure

2.11.1. A more durable fabric sleeve that holds all the components in place and in the right position. This is how the user will physically interact with the device.

2.12. Tolerance Analysis

2.12.1. If the voltage regulator is not able to keep the voltage within our tolerance of voltage then the devices could become damaged or (in the case of the pressure sensors) give very varying results. We can possibly mitigate the issue with the pressure sensor if we were to measure the voltage digitally with the microcontroller. This will be especially useful during testing so we can find where an acceptable current limiting value is.The maximum the voltage can vary is 0.5 volts due to the distance sensor only allowing voltage values of 5 +/- 0.5 volts. Our arduino substitute should be able to measure the difference between the ½ kilogram tolerance we set. The sensor variable resistor changes from 0 to 1000 with masses applied of 50 kg to 0 kg respectively. This means the sensor has a rate change of 1000/50 Ohm/kg. With our tolerance of ½ kg this leads to 10 ohm changes. The entire system consists of a voltage divider with another static 1000 ohm resistor. This means the system needs to read 5 v*10 ohm/(1000 ohm + 10 ohm) = ~0.05 v. An arduino can measure up to 5 volts with 1024 distinct values. In other words the arduino can differentiate up to 5v/1024 = ~0.005v, which is more than acceptable.

Table 1. Block Diagram Requirements and Verifications

Block Diagram Item	**Requirements**	**Verifications**
Pressure Sensors - These devices measure what force the user is applying to cushion and send the data to the microcontroller.	The pressure sensor must be able to differentiate to a degree of at least ½ a kilogram.	Start recording pressure data with no load, set this value as the zero. Then add ½ kilogram weight, set this value as ½ kilogram in the system. Increment weights of ½ kilogram and make sure that the data corresponds to accurate values.
Battery - Supplies power to the device.	Supply current equal to at least 2.0 A for all the components.	Attach a set of power resistors and measure the current through the system with an ammeter. The battery should be able to sustain at least 2.0 A.
Voltage Regulators - These/this converts the battery voltage to the correct level for each component.	Should supply 5 volts ± 0.1 volts. Should be able to supply 2.0 A ± 0.1 A for the microcontroller and 10 mA ± 1 mA	Run two tests per regulator. The first test will use a bench power supply at battery nominal voltage. The output should match what is required. Then use the battery as the supply and measure the voltage with a changing load (i.e. add a sensor or two) and verify the voltage matches what is requested.
Microcontroller - Interprets data from pressure sensors and distance sensors and sends the raw data to the phone application via a bluetooth module.	This should send the data to the bluetooth module at least 4 times a second.	Power the microcontroller and upload the code to the chip. Use an oscilloscope and determine that the pin that sends data to the Bluetooth module sends data 4 times per second.

Bluetooth Module - The bluetooth module should transmit data between the microcontroller and the phone app.	This should be able to send off the data from the microcontroller at least 4 times a second.	With the Bluetooth module powered and receiving data from the microcontroller, verify on the phone that the data matches the required speed.
Phone App - The phone app serves as a user interface. It should allow input and storage of baseline data from the microcontroller for postures the user considers as healthy. It will also evaluate data sent from the microcontroller via bluetooth and warn the user of bad postures in real time.	1. The app should detect if it is run on a compatible platform in terms of hardware, operating system, and library support. 2. It should provide the user with an “Set Default” mode that allows input and storage of a set of baseline data. 3. It should be able to process all the incoming data, form a weight distribution map, determine if the user is leaning towards any direction, and give visual feedback to the user. This should all happen and update 4 times a second.	1. Run the app on an emulator with different operating system versions and libraries. Verify that the software only generates an error message when its requirements are not met. 2. Run the app on an emulator with a hardcoded set of inputs under the “Set Default” mode. Verify that the data can be written and read correctly. 3. Run the app on an emulator with a hardcoded stream of inputs which simulate sensor data. Update the inputs 4 times every second during the simulation. Record and display the app’s output history. Verify that the phone app has generated correct visual feedback for every input and that no input is missing from the display.

Distance sensor - This device is mounted towards where the upper back of the user rests. It measures the distance of the upper back and helps determine if the user is in a compromising position.	This device must be accurate to within 1 cm.	Power the sensor and place reflective material at various distances, each 1 cm apart. Determine the data matches the actual distances.

3. Project Differences

3.1. **Overview**

3.1.1. Similar to the SP15 COMPACT SLOUCH DETECTOR and SP18 ORTHOPEDIC CHAIR, we plan on monitoring the sitting position of the user and telling them how they could improve their posture.

3.1.2. The SP15 project designed a wearable suit that aims to correct posture by isolating muscle groups, applying feedback to the muscle group directly responsible for a slouch, and rewarding good postures. This design provides a solution that tries to develop muscle memory of good posture for the user, and requires a suit with components dedicated to each muscle group to work. The suit uses a timer for posture correction sessions, and will require consistent use for the user to fully utilize its function.

3.1.3. The SP18 project aims to solve the same problem by using a chair with embedded sensors to detect bad postures. The sensors are placed beneath the seat and at the back of the chair. A microcontroller is responsible for sending sensor data to a computer via Bluetooth for processing, after which the chair uses vibration to warn the user of bad postures. To make most use of this design, the user needs to make sure they always have access to a computer and must carry the chair around when working at different locations.

3.1.4. Our proposal is a combination of the two mentioned projects that covers both the sitting and back posture, and would use a phone app component for checking if the posture is good and providing feedback. This feature, plus that our design is a cushion instead of an entire chair, makes our design portable and could be placed on any seat that one would be sitting on, minimizing the inconvenience caused by the user having to always use a fixed chair or put on a wearable device, which is the case with the previous projects.

3.1.5. List of design differences between our project and the SP15 project:

3.1.5.1. The main difference from the 2015 project is the removability of the product from the user. Our project is not worn but rather placed on a seat and is easily removable, which makes it a more convenient choice. This change makes our design less accurate because we are not adhered to the user, but it still gives us accurate enough data to help with posture.

3.1.5.2. We do not have haptic feedback like the 2015 project but rather a visual feedback system based upon our app. The issue with this change in design is that the system is a little more complicated i.e. there are more things that could go wrong. The benefit is that the physical pieces will be cheaper because we do not need to add vibrating parts, and the app allows the user to track and analyse their own data in real time or over an extended period of time.

3.1.5.3. Instead of being used as a tool in timed training sessions like the SP15 design, our design provides short-term feedback that accompanies most of the user's sitting time. As a tradeoff, our design does not have the long-term benefit of slowly developing muscle memory of good postures for the user, but instead it aims to constantly identify and warn the user of bad postures as much as possible.

3.1.6. List of design differences between our project and the SP18 project:

3.1.6.1. Improved mobility due to the cushion being portable. Trade off: light and easily portable fabric enclosure that we need to use may not be durable enough to protect embedded components in the long term compared to a solid chair.

3.1.6.2. Our phone app, compared to the Posture Analysis GUI used by the SP18 project, offers a more easily personalized analysis as it allows the user to set their own baseline data that they consider to be healthy, instead of relying on a series of calibrations that will likely only make small adjustments based on machine learning, which was the solution of the SP18 project. Trade off: letting users set their own standard is more tolerant to bad postures if the users are not aware that they have already formed a potentially bad posture while sitting normally. Thus, our analysis can be less accurate for users with bad sitting habits.

3.1.6.3. We use a phone app as our user interface that is more accessible than a computer software, which the SP18 project relied on. This matches our idea to make the project more portable. Tradeoff: due to a phone having limited computing power compared to a computer, we removed the potentially demanding machine-learning-based calibrations and introduced a different approach towards customization, as mentioned by 5.1.6.2..

3.1.6.4. Same as 5.1.5.2., we use visual feedback from our phone app instead of sending vibrations to the user.

3.2. Analysis

3.2.1. Compared to the SP15 project, the usage of our design is more flexible. The user does not need to put on a sensor-embedded suit and divide their time into training and resting sessions, which may cause troubles for the user's time management since it requires the user to remain using the device for a sustained period of time. For people who are only able to sit still for short or unpredictable periods of time, the SP15 design may not work well with their schedule. With our design, however, the user can place the device on wherever they sit and sit for however long they want.

3.2.2. Compared to the SP18 project, our design is easily portable, and handles customization in a more straightforward way.

We applied our design on a cushion with a phone-based interface instead of on a chair with a computer-based interface, which made sure that our device can be easily carried around. This solves a problem in the original SP18 project that caused the device to practically be only usable when the user is at home.

Furthermore, instead of relying on machine learning for calibration, our design will directly allow the user to set their own baseline weight distribution while sitting on the cushion. There are several reasons behind this change:

First, a good machine learning algorithm will need to adjust its weight parameters based on a large pool of samples, which is hardly available for our project. Without enough samples, the machine learning algorithm will essentially be untested and thus unreliable, which will most likely generate data that is a non-scientific mix of the user's weight distribution ratio and the default value. In this regard, our data can be more reliable out of the box.

Secondly, machine learning may have a hardware requirement that exceeds certain mobile phones' capacity. To keep our idea that the entire device should be easily portable and accessible, we decide to make the software less demanding.

Furthermore, our implementation is more transparent and thus is easier to understand for an average user. The SP18 project ended up using a classifier to only inform the user of whether their current posture is "good" or "bad". This feedback is ambiguous and does not provide enough information regarding how to improve the posture. Our project uses the ratio of the user's sitting weight distribution calculated from sensor data and can thus inform the user of what direction they are leaning towards.

So in conclusion, our design is generally more accessible and convenient compared to the SP18 project.

4. Cost and Schedule

4.1. Cost Analysis

4.1.1. Labor: (For each partner in the project)
$50/hour x 2.5 x 50 = $6250 TOTAL per person
Labor For Three Person Team = 3 * TOTAL per person = $18750

4.1.2. PCB Print: Bay Area Circuits.
$75 TOTAL per board
Need 3 boards for testing = $225

Table 2. Part Costs

Description	Manufacturer	Part #	quantity	Unit Cost
Pressure Sensor	SparkFun	SEN-10245	8	10.95
Battery (LiPo)	Goldbat	Amazon Link	1	16.99
Voltage Regulator	NTE Electronics	VUPN7407	4	6.95
Arduino	MegaChip	ATmega2560	1	11.99
Bluetooth Module	Silicone Labs	BLE112-A-V1	1	11.26
Distance Sensor	Sharp	GP2Y0A21YK0F	2	9.99
Pads (Upholstery)	FoamTouch	Amazon Link	1 Roll	33.93
Fabric Covering	FabricLA	Amazon Link	2 x 1 yard	14.65

4.1.3. Sum of costs into a grand total: $330 + $18750 = ~$19080

4.2. **Schedule**

Week 1	Design and order Microcontroller board (C, D)	Order Unit test hardware (D)	Pick and order Components (C, F)	Make a Bluetooth-capable app (F)
Week 2	Construct board and physical parts(C, D)		Unit tests and calibration (C, D)	Design front-end and data analysis for app (F)
Week 3	Module Tests (C, D, F)	Full design tests (C, D, F)		

5. Discussion of Ethics and Safety

5.1. **Personal Care**

5.1.1. Due to the fact that a human will literally be sitting on an electrical system strong care must be taken to ensure maximum safety.

The main way this problem is tackled is by separating the user from the electrical components with a layer of insulating fabric and an insulating foam pad (in addition to its qualities in making the apparatus more comfortable). The wiring should be contained as much as possible within the fabric sleeve to distance the user.

Luckily, the main parts that the user will physically interact with are the pressure sensors and the distance sensor, as the remaining components drape behind the seat or are based on a phone. The two sensors will require very little power, but they will have up to 5 volts across so we will attempt to limit the current able to be supplied by tuning the voltage regulator related to the sensors to only about 10 mA. This is still a dangerous current amount but the lower we are able to tune the system the better.

These precautions should comply with the IEEE Code of Ethics #1 which states that we will agree “to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment” [3]. In fact our product may even increase the health of the public as back problems are very detrimental to our health for more than one reason [2].

5.2. **Health Guarantee**

5.2.1. We cannot say, though, that our product is guaranteed to help the health of the user. This is because good posture will not *absolutely* help one’s back and we cannot ensure that the user will use the product to the extent one might need. We do not guarantee this statement because we try to adhere to IEEE Code of Ethics #3 which states that we agree “to be honest and realistic in stating claims or estimates based on available data” [3]. We do state that proper use of our product will help a sizable margin of the moderate or less severe posture cases, but not all cases.

Our application setup requires the user to sit in a position that they determine to be “proper” posture. This is because each person is unique in their sitting pattern and potentially what weight distribution they possess. This complies with IEEE Code of Ethics #8 that says “to treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression” [3]. We do not intend to discriminate if the user has a weight distribution that does not align with our original testing or code for whatever that reason may be.

6. References

[1] “Chronic Back Pain,” *Health Policy Institute*. [Online]. Available: https://hpi.georgetown.edu/backpain/. [Accessed: 04-Apr-2020].

[2] J. Huizen, “Tips for Sitting Posture at a Computer,” *Medical News Today*, 21-May-2018. .

[3] “IEEE Code of Ethics,” *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 04-Apr-2020].

[4] “UNITED STATES DEPARTMENT OF LABOR,” *eTools | Computer Workstations eTool - Good Working Positions | Occupational Safety and Health Administration*. [Online]. Available: https://www.osha.gov/SLTC/etools/computerworkstations/positions.html. [Accessed: 04-Apr-2020].

[5] “Arduino Mega 2560 Rev3,” Arduino Mega 2560 Rev3 | Arduino Official Store. [Online]. Available: https://store.arduino.cc/usa/mega-2560-r3. [Accessed: 18-Apr-2020].