Back to Healthy Posture

Design Document

Daniel Chen, Fangqi Han, Christian Held
Group 10

ECE 445

April 17 2020

Table of Contents

1. Abstract

Good sitting posture is something to strive for, as there are numerous health benefits from reduced lower back pain to increased circulation to other health benefits. However, monitoring oneself over the course of a work or study day can prove difficult, as people tend to return to their usual sitting posture over time without oversight. There are also more components of bad posture that can be measured, such as lack of movement and slouching, imbalanced sitting posture. Similar to the SP15 COMPACT SLOUCH DETECTOR, the SP18 ORTHOPEDIC CHAIR, and the SP20 POSTURE-SENSING SMART CHAIR we plan on monitoring the sitting position of the user and telling them how they could improve their posture.

The SP15 project was a portable suit that aimed to correct posture by applying vibrational feedback to the muscles directly. The SP18 and SP20 had the same design of a chair that also senses a slouch in the user and tells the user by either vibration or a set of LEDs. This project tries to bridge the gap between a portable and comfort between these two designs by implementing a pad that lays on top of a chair and uses both pressure sensors and distance sensors. The feedback system is directed to the user by an Android application that will process the data into information related to the user's posture from how they apply pressure to the bottom and backrest of the seat as well as any slouching. This differs from the SP18 and SP20 projects that had a classifier that only categorizes "good" and "bad" posture, and they would return this information as haptic feedback with those vibrations or LEDs rather than telling the user how to improve their posture.

2. Second Project Motivation

2.1. Updated Problem Statement

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].

2.2. Updated Solution

The system will consist of 8 pressure sensors, 2 distance sensors, 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. This evaluation process will be handled by the phone app once raw data is sent from the microcontroller via Bluetooth to the user's phone.

2.2.1. Backrest Design

One distance sensor will be mounted toward each of the user's shoulders. The difference between the two readings will be evaluated by the phone to warn the user if one of their shoulders is leaning forward.
Furthermore, the average of the two distance sensors will be used as an estimate of where the user's back is. This will be analyzed together with pressure sensor readings from the seat to evaluate if the user is leaning forward. More details will be introduced in part 1.2.3..
For a healthy posture, the user will be expected to sit upright and not have their back leaning against the back of the chair. Thus 2 pressure sensors are also placed at different heights at the backrest. Once sensed pressure from the user, their readings will be used by the phone app to warn the user against leaning towards the back of the seat. This feature may be strict for some users, so it can be switched off in the phone app.

2.2.2. Seat Design

6 pressure sensors are placed on a soft pad on the seat of the cushion, with 3 sensors under each leg. Data is sent from these sensors to the phone app to form a weight distribution map.

	Left Leg	Right Leg
Front	Pressure Sensor 1	Pressure Sensor 2

Middle	Pressure Sensor 3	Pressure Sensor 4
Rear	Pressure Sensor 5	Pressure Sensor 6

FIgure 1. Table of Weight Distribution Map

The app will compute what ratio of the user’s pressure on the seat is distributed to each sensor. This ratio will be used to evaluate the user’s posture.

Note that the user may not sit on every sensor. Specifically, the user may need to move forward in order to reach out to a desk while still keeping their back straight. Thus pressure sensors 5 and 6 at the rear may not receive any pressure from the user when they move forward. However, this does not imply a bad posture. Therefore, we will use the rear-most “pressured” sensors that do sense the user’s weight as an implication for the phone app to calculate where the user’s back should be. This will be compared to the distance data sent from backrest, which indicates where the user’s back really is.

2.2.3. Warning Types

The phone app will send visual warnings to the user based on evaluations of pressure and distance data sent from the microcontroller.

A left/right-leaning warning will be sent when distance data from the two shoulders have a difference larger than 5 cm, or when pressure ratio under the two legs have a difference larger than 15%. Both of these thresholds can be and are recommended to be calibrated by the user to fit their personal preference.

A forward-leaning warning will be sent when the distance data gives a distance 5 cm longer than the estimated value based on data from the rear-most “pressured” sensor. The forward-leaning warning will also be sent if the 2 rear-most “pressured” sensors do not take up at least 50% of the user’s pressure on all on-seat sensors. This is because we expect the trunk of an average person to take up at least 50% of their weight, and when a person is sitting upright, their back should generally be aligned with their rear without leaning forward too much. Again, these thresholds can be set by the user.

A “please do not lean on the backrest” message will be sent when the pressure sensors at the back are pressured and this warning is not switched off.

A “please sit on the soft pad instead of on the edge of the chair” message will be sent if no pressure sensor or only the front-most ones detect the user’s weight. We assume the user is trying to use the device if the cushion is powered and the phone app is opened.

2.2.4. Calibrations

We will allow the user to set their own thresholds in the phone app. Alternatively, there will be a “set default” mode where the phone app asks the user to sit in a posture that they consider as healthy. The raw sensor data from the microcontroller will be compared with the default thresholds and generate offsets for each of them to be adjusted.

2.3. Updated High-Level Requirements

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

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

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

2.4. Updated Visual Aid

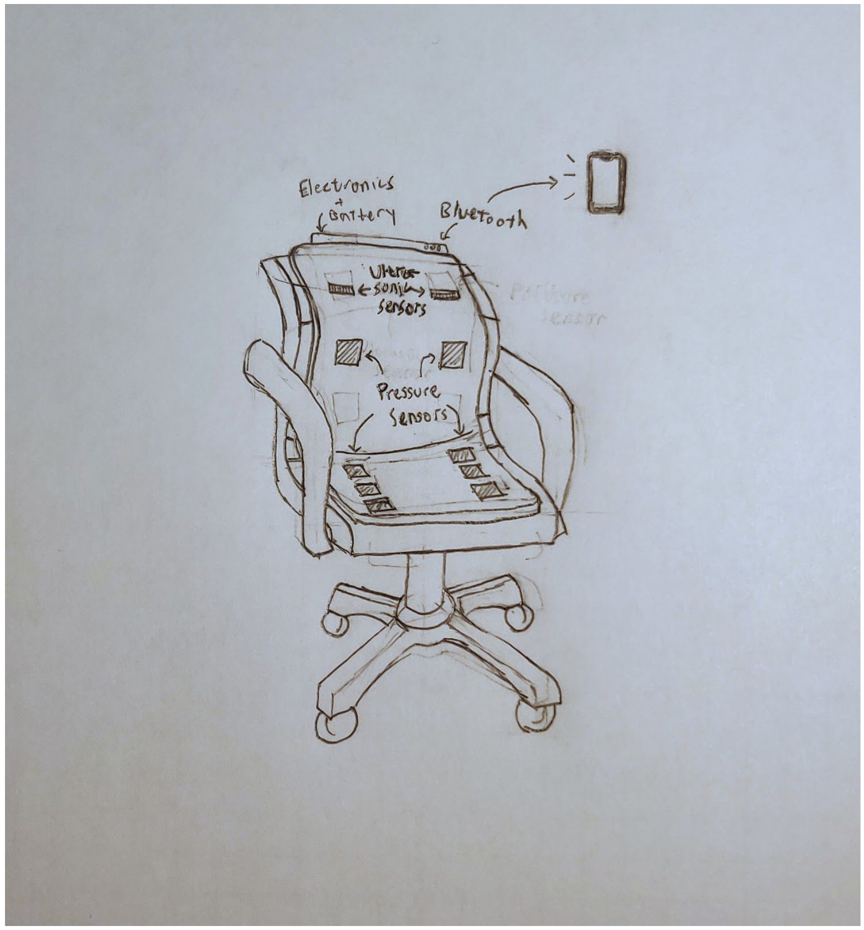

Figure 2. Visual Aid

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.

2.5. Updated Block Diagram

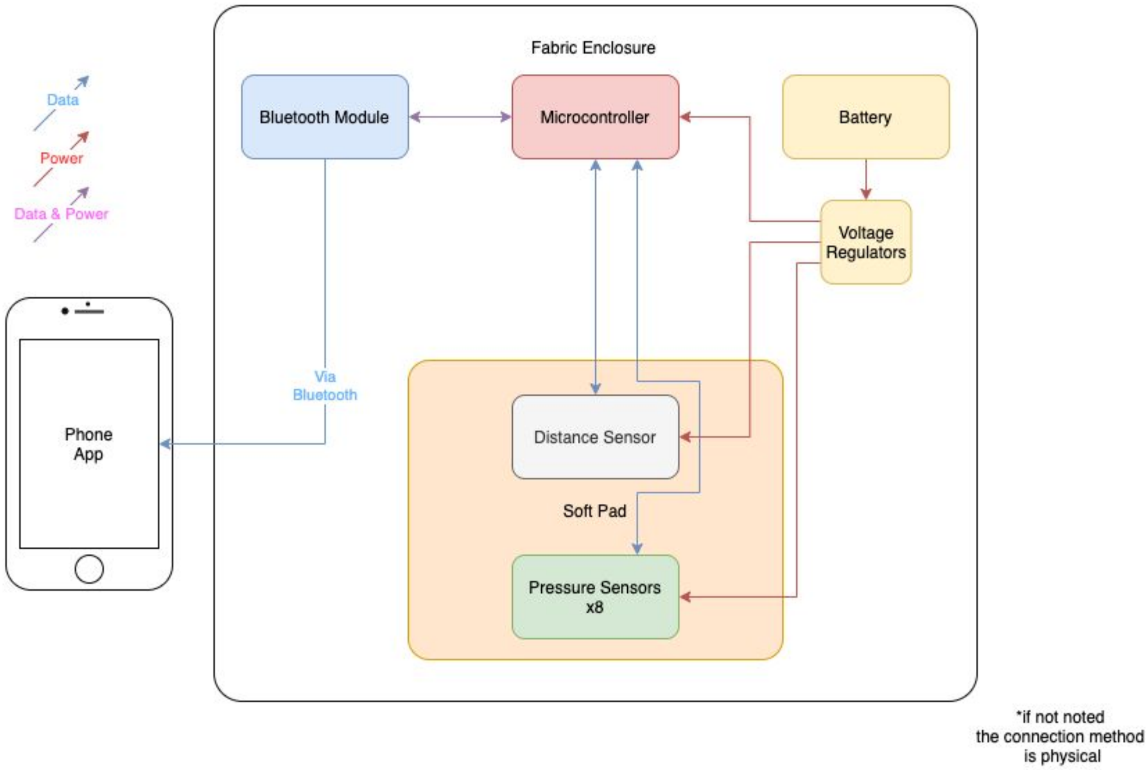

Figure 3. Block Diagram

After the battery voltage is regulated the microcontroller reads the data from the eight pressure sensors and two distance sensors and then 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.

3. Second Project Implementation

Our design is mainly based upon things we would need to physically construct with tools to which we do not entirely have access. Since we are unable to do work in the lab, a lot of the project analysis is based upon hypothetical problems we could run into. Finally, it is necessary to explain how one would use the device because no demonstration is incorporated this year.

3.1. Implementation Details and Analysis

3.1.1. User device instructions:

3.1.1.1. Download the Android application.
3.1.1.2. Plug in the battery
3.1.1.3. Connect Android application to the sensing pad
3.1.1.4. Place pad on the desired chair
3.1.1.5. Sit in healthy[1] and balanced posture
3.1.1.6. Go to reset function on the application
3.1.1.7. Hold the position until application has reset itself
3.1.1.8. When the device detects that the user has started to hold a position that would be considered unhealthy the phone application alerts them to sit in a healthier position. There would be a heat map of each pressure pad (on a model of a human sitting). Additionally, percentages of how much the user has deviated from what they set as healthy.
3.1.1.9. At random intervals between 10-15 minutes the device alerts the user to hold a different healthy position so as to not give discomfort to the user.

---

[1] There would be additional documentation to teach the user how to sit properly. As explained by the Occupational Safety and Health Administration [4] "

- *Hands*, *wrists*, and *forearms* are straight, in-line and roughly parallel to the floor.
- *Head* is level, forward facing, and balanced. Generally it is in-line with the *torso*.
- *Shoulders* are relaxed and *upper arms* hang normally at the side of the body.
- *Elbows* stay in close to the body and are bent between 90 and 120 degrees.
- *Feet* are fully supported by the floor or a footrest may be used if the desk height is not adjustable.
- *Back* is fully supported with appropriate lumbar support when sitting vertical or leaning back slightly.
- *Thighs* and *hips* are supported and generally parallel to the floor.
- *Knees* are about the same height as the hips with the *feet* slightly forward"

3.1.2. Schematic

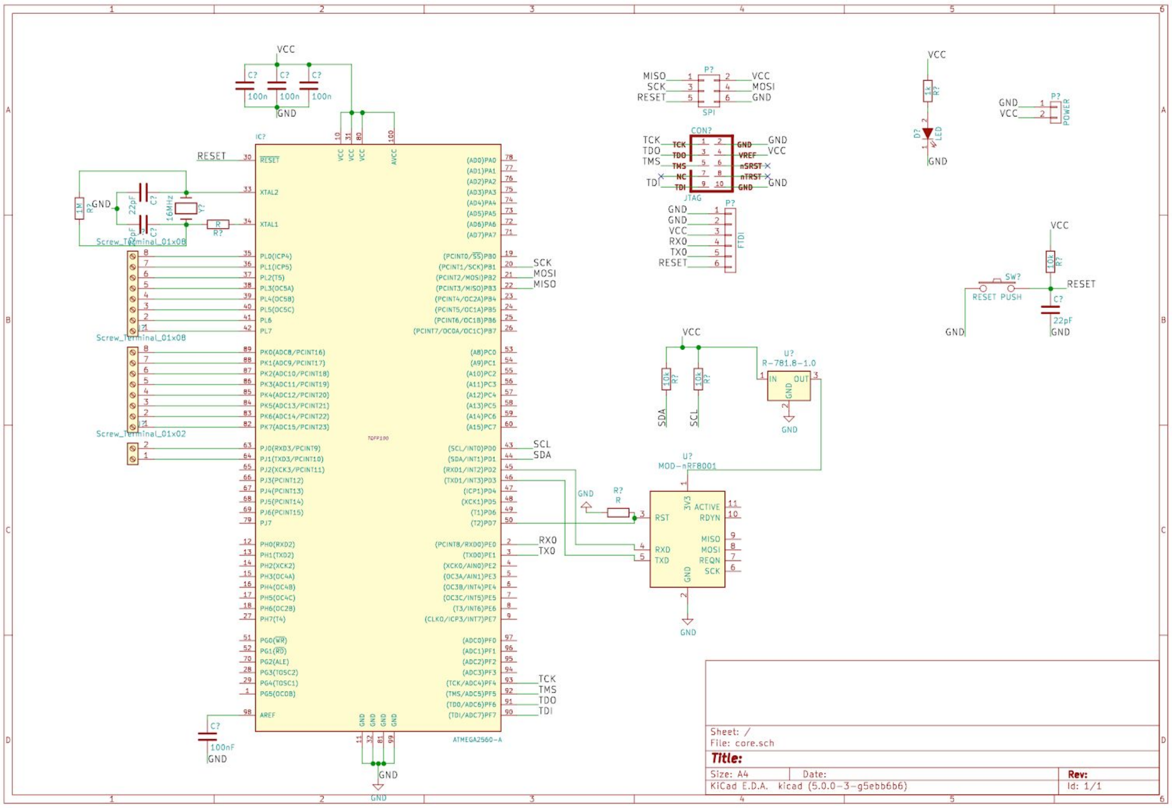

Figure 4. Arduino Schematic

Figure 4 shows the used schematic based upon 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. The things added are the terminals for the nine sensors (shown on the left of the main integrated circuit (IC), the large yellow box), the power conversion and the serial connections to the bluetooth module (the smaller yellow boxes, small to medium respectively). The other circuitry is just to make the IC run with base functions, such as power, clock circuitry, reset switches, indicator LEDs, and I^2C protocol pull up resistors.

3.1.3. Pressure Sensors

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.

3.1.4. Battery

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.

3.1.5. Voltage Regulators

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.

3.1.6. Microcontroller

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.

3.1.7. Bluetooth Module

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.

3.1.8. 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, as mentioned before in section 1.2.4.. It will also evaluate data sent from the microcontroller via bluetooth and warn the user of bad postures in real time. The algorithm used is introduced in section 1.2.. Basically, the phone app will do the calculations to determine what is a good posture. It is recommended that the user go through the process of setting up the application each sitting session due to chair differences. This also accounts for individuals who deviate from the “normal” sitting position (due to an amputation, bone defect, etc.).

3.1.9. 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. The device uses an ultrasound sensor that emits a sound wave and then listens for the sound to be reflected back. It then calculates the time it took for the sound to come back to determine distance.

3.1.10. Soft Pad

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.

3.1.11. Fabric Enclosure

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.

3.1.12. Tolerance Analysis

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.

4. Second Project Conclusions

4.1. Implementation Summary

As we were a mainly hardware based project, there is not a vast amount of work that could have been done, given the situation with COVID-19 and the loss of lab access. What we were able to implement was the creating the schematic for the circuitry of the logic of the project and

4.2. Unknowns, uncertainties, testing needed.

We have yet to physically construct most of the device and so listing each part would not be beneficial for analysis. Nonetheless, there are some things that would need to be tweaked as we constructed the sensing pad. We would need to spend some time calibrating each pressure sensor, as each part and resistor set up has some variance to the system. It would be necessary to find the padding thickness that is comfortable. That measurement is an opinion based decision so actual testing is required in order to be sure that a reasonable comfort level is achieved.

4.3. Ethics and Safety

4.3.1. Personal Care

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 sensors, 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].

4.3.2. Health Guarantee

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.

4.4. Project Improvements

4.4.1. The battery is a point of contention; it is quite dangerous. A very strong enclosure is required. Perhaps making the device able to run off of rechargeable AAs would be beneficial for many reasons, the first being safety. Additionally, one can swap batteries and theoretically, with enough batteries being cycled, one can use the device indefinitely. This does lead to more waste products, especially if the user does not use rechargeable batteries.

4.4.2. If the device had a screen attached somehow, then we would circumnavigate the issue that not all devices can run Android applications. The calculations our device does are mostly linear and can definitely be run on a microcontroller. The user then does not also need to have a “smart” device, and we can remove the bluetooth module from our design.

4.4.3. Furthermore, more time would allow us to test out our algorithm for posture detection. Our implementation is largely based on our own understanding of common unhealthy postures and cannot be tested with enough samples due to limited resources during development. It might also be useful to pour many more hours on our visual feedback of said algorithm. Being able to give a better visualized message of what the device is interpreting to the user would be beneficial for data communication.

5. Progress made on First Project

Before we moved on to the second project, we were able to design the main PCB and a firmware structure used to manage keyboard layers.

5.1. PCB file

The main board for the keyboard was nearly at completion. The only remaining feat was to finish routing all the wiring, and this was partially outlined as we placed each component (the hard work was completed). This included meticulously creating a grid for each of the individual keys to reside. Also we had to reference each individual key to a precise size so it would fit according to the standardized keyboard layout.

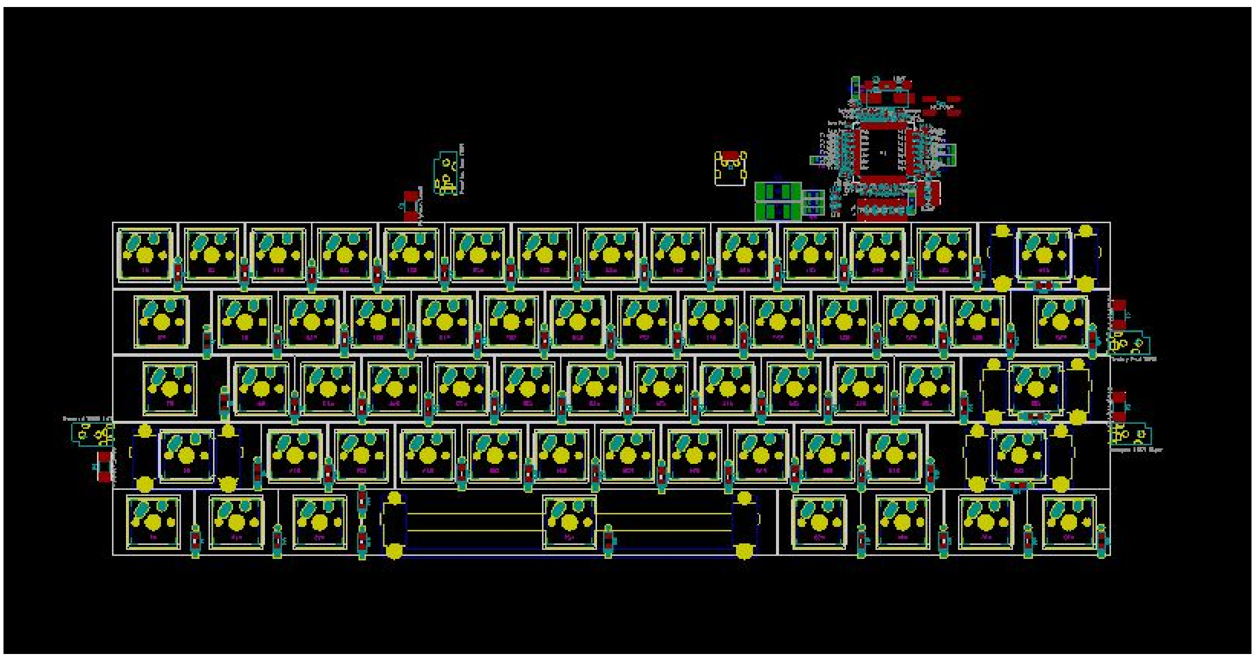

Figure 5. Project 1 PCB file
This is the nearly-finished PCB file from our first project. The only remaining steps left would be to route all the connections then send the gerber files to be printed.

5.2. Layer Manager Array

This is a structure in the firmware implemented to solve the “Fall Through” case with layers, and can also be used to handle “sticky” layers.

To explain in short, sticky layers are layers that stay active once activated by a keypress. These layers only switch off upon a “deactivate” command (e.g. CapsLock on a typical keyboard). “Fall through” cases are where the most recently activated layer does not have a keycode assigned to the currently pressed key. In this case, the keyboard falls through recent layers until a keycode corresponding to the pressed key is found on a not-so-recent active layer and returns that keycode. We need to keep track of the orders of layer activation to determine which layer the keyboard should “fall through” to.

The structure we made records the following for each active layer:

- layer_num is a number that matches the newly activated layer. Each layer designed in the keyboard layout will have a unique layer_num assigned to it. This number identifies what layer in the layout that an activated layer is (i.e. identify if it is a layer activated by Shift or Alt by design)
- is_sticky evaluates if the layer is sticky and will be used to handle sticky layers.
- stack_id keeps track of the order that the layers were activated. Layer_used records if a stack_id is occupied. Each active layer will have a unique stack_id.
- stack_top points to the current top of the array. Note that the array structure is not really a stack, as the user’s inputs are unpredictable and can press/release keys to pop layers from anywhere in the array. stack_top is only needed for storing new layers. A loop must be used during popping layers, where stack_top will be decremented afterwards.

Pseudocode for recording a new layer with the layer manager array structure:

```
for (int stack_id = 1; stack_id > 0; stack_id++) {
   if (layer_used[stack_id] == false) {
      layer_used[stack_id] = true;
      stack_top++;
      # Record the new layer’s attribute values in the array
      layer[stack_top].layer_num = layer_num;
      layer[stack_top].stack_id = stack_id;
      layer[stack_top].is_sticky = is_sticky;
      return;
   }
}
```

6. References (graded with English and formatting)

[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].