# Individual Technical Journal

Senior Design

Slate Jordan

# Week of September 5$^{th}$-11$^{th}$

### Summary

This week our team struggled to find an idea that met all our criteria. We tentatively decided on my idea for an automatic pothole reporting device, however, more research yielded AI with cameras to be the best approach which doesn't give our EE team members much hardware work. Considering this we switched to our backup Evergy sponsored idea of QR Code labeling and database integration. Our concerns about not enough work were verified in class and we were back at square one with no project ideas left. Eventually we settled upon an idea from our meeting with the professor.

### Focus of the week

This week involved lots of research on various ideas we had. The main problem with my first idea was the work involved for a solution being mostly software which isn't ideal for our team. Our backup idea was never ideal and was mainly used as a placeholder after we ran out of time between nixing the first idea and presenting on Friday. The current plan is to go with a local license plate scanner and reader which uses the LoRaWAN network to communicate findings back to police stations. Currently I am researching existing solutions commonly called LPR or license plate readers to find out if our solution can be unique enough to differentiate it from existing solutions.

# Week of September $12^{th}$-$18^{th}$

**Summary**

We were still struggling with our project idea this week, so the majority of my time was spend researching and thinking about projects. We had multiple team meetings and communication about the validity of ideas. The ALPR (Automatic License Plate Reader) idea was slowly falling apart after much research into existing technologies revealed many solutions already on the market with capabilities that met our requirements.

**Research**

My research led me to various existing ALPR devices and software. I will use platerecognizer.com as an example. It supports both on-device and cloud software, images, and video, as well as non-ideal angles, lighting, resolution, etc. With all this being considered I concluded that anything we would be able to build with our limited resources and time would not have any competitive advantage with the main aim of our ALPR solution being the ability to read plates at angles.

**Some Resources**

https://github.com/openalpr/openalpr

https://www.openalpr.com/

https://platerecognizer.com/

# Week of September 19th-25th

**Summary**

Our team was still toying with a few ideas like building a map of traffic and speed cameras using computer vision to give drivers the knowledge of where these controversial traffic cameras are located. Eventually we settled on a keyboard with the ability to detect finger placement. This would in theory speed up the process of learning how to touch type. Our initial idea to enable the sensing of finger placement is capacitively sensitive keycaps. We would also have software to give an on-screen representation of finger placement thus discouraging the habit of looking down at the keyboard. The problem: some people have trouble learning touch typing and breaking the habit of constantly looking down at the keyboard. The solution: A keyboard with touch sensitive keys and an on-screen overlay showing which keys are being touched.

**Research/Thoughts**

My initial idea on how to sense key touches was to have the keys be capacitive. I thought of the clever Oculus Rift controllers which have capacitively sensitive buttons and triggers to show in a virtual environment where your fingers were currently resting. As is the case with many technologies, the original purpose is not always the best use so with that in mind I am still brainstorming other possible use cases for this extra layer of input being enabled on a keyboard. Another idea for the tracking the position of fingers above the keyboard was some type of sensor(s) to track the x-y position, but we think capacitive keys is the clear first step with that being something to consider later in the development process. Using a keyboard with standard Cherry MX style key caps will allow us to switch keycaps out as we go forward with testing.

**Resources**

https://www.youtube.com/watch?v=wdgULBpRoXk

https://www.youtube.com/watch?v=mWR9Q_pTagw

https://www.youtube.com/watch?v=bokWXzVXsyc

https://www.nelson-miller.com/capacitive-switch-work/

http://www.pcbheaven.com/wikipages/How_a_Touch_Button_works/?p=1

https://butlertechnologies.com/basics-capacitive-touch/

# Week of September 26th- October 2nd

**Summary**

My focus for this week was to find out as much as I could about affordable and small capacitive sensing hardware. Something small enough we could incorporate into a keyboard as well as interface with windows via USB.

**Research Findings**

An Arduino stood out as the clear choice for microcontroller for all our capacitive inputs and interfacing with the computer. A combination of the broad userbase and resources and hardware addons available for Arduinos make for a nice fit. Specifically, the Arduino nano model is ideal for its small form factor. For the capacitive sensing of so many keys it is impractical to directly attach them to an Arduino. This means some intermediary controller/input device is needed to take all 18 or so inputs. The clear choice for this was the Adafruit 12-Key Capacitive Touch Sensor Breakout - MPR121 for it's 12 capacitive inputs as well as built in sensitivity adjustment and noise filtering for only $8.

**Resources**

https://www.adafruit.com/product/1982

https://store.arduino.cc/products/arduino-nano

https://create.arduino.cc/projecthub/gatoninja236/capacitive-touch-sensing-grid-f98144

# Week of October 3$^{nd}$- 9$^{th}$

**Summary**

I didn't do much in the way of technical research this week with our midterm presentation coming up. I helped work on the presentation as well as brainstorm use cases for our keyboard as well as some routes we could take with the software.

**Resources**

https://windowtop.info/

https://docs.oracle.com/cd/E19683-01/816-0279/transpoverlay-63475/index.html

https://polycount.com/discussion/89191/image-always-on-top-overlay-software-for-win

https://softwarerecs.stackexchange.com/questions/1767/display-half-transparent-ghost-image-over-other-windows

https://www.mobzystems.com/tools/seethroughwindows/

# Week of October $10^{th}$- $16^{th}$

**Summary**

This week we continued the theme of brainstorming and researching problems or demographics of people which could benefit from our keyboard. I also researched and found a keyboard which met our requirements of price, size, switches, and quality. In addition, I had a breakthrough idea on how to easily operate the Arduino USB connection over the same keyboard cable.

**Ideas**

In my research on how to easily operate two USB connections over the existing keyboard cable without interfering with standard keyboard operations, I had an idea to easily overcome the challenge. After seeing keyboards with built in USB pass through ports, It was clear that we could start with a keyboard that had a USB pass through port built in. This would allow us to easily tap into the pass through and connect our Arduino via USB with no changes to the existing keyboard design or function.

# Week of October 17$^{th}$- 23$^{rd}$

**Summary**

My focus for this week was mostly on hardware and getting more specific requirements and specifications. I've made the selection of silicone insulated wire for our keyboard wiring due to its high flexibility which should minimize interference with keys. From looking over our keyboard, space inside the chassis is limited so my initial thoughts are we might have to find a different keyboard or modify the bottom plate to make room for our parts. Our team's technical report was also a big focus for this week and incorporated the research done.

**Resources**

| Amperes | 0-4 ft. | 4-7 ft. | 7-10 ft. | 10-13 ft. | 13-16 ft. | 16-19 ft. | 19-22 |
|---|---|---|---|---|---|---|---|
| 250-300 | 4-ga. | 2-ga. | 2-ga. | 1/0-ga. | 1/0-ga. | 1/0-ga. | 2/0-ga. |
| 200-250 | 4-ga. | 4-ga. | 2-ga. | 2-ga. | 1/0-ga. | 1/0-ga. | 1/0-ga. |
| 150-200 | 6 or 4-ga. | 4-ga. | 4-ga. | 2-ga. | 2-ga. | 1/0-ga. | 1/0-ga. |
| 125-150 | 8-ga. | 6 or 4-ga. | 4-ga. | 4-ga. | 2-ga. | 2-ga. | 2-ga. |
| 105-125 | 8-ga. | 8-ga. | 6 or 4-ga. | 4-ga. | 4-ga. | 4-ga. | 2-ga. |
| 85-105 | 8-ga. | 8-ga. | 6 or 4-ga. | 4-ga. | 4-ga. | 4-ga. | 4-ga. |
| 65-85 | 10-ga. | 8-ga. | 8-ga. | 6 or 4-ga. | 4-ga. | 4-ga. | 4-ga. |
| 50-65 | 10-ga. | 10-ga. | 8-ga. | 8-ga. | 6 or 4-ga. | 6 or 4-ga. | 4-ga. |
| 35-50 | 10-ga. | 10-ga. | 10-ga. | 8-ga. | 8-ga. | 8-ga. | 6 or 4-ga. |
| 20-35 | 12-ga. | 10-ga. | 10-ga. | 10-ga. | 10-ga. | 8-ga. | 8-ga. |
| 0-20 | 12-ga. | 12-ga. | 12-ga. | 12-ga. | 10-ga. | 10-ga. | 10-ga. |

Length in feet

https://www.ql-custom.com/what-is-silicone-wire-performance-benefits-and-uses-with-examples.html

https://www.meridiancableassemblies.com/2021/04/wire-gauge-size-guide/

# Week of October 24$^{th}$- 30$^{th}$

Continued the planning of our prototype. Also continued working on other possible uses for the extra layer of input enabled by our capacitively sensitive keys. We continue to have the problem of what the best use case or use market is. One option to enhance the touch typing learning experience with our keyboard is to detect which finger is on a certain key. The ability to distinguish between fingers will allow our program to analyze one's typing habits and report on how to improve form and in turn increase typing efficiency and speed. This might be enabled by doing an initial calibration for an individual user's fingers if our capacitive system is sensitive enough to detect the subtle change in capacitances. Another option to enhance this capability would be to incorporate a camera which could augment the capacitive sensors in distinguishing between fingers. VR application is already using a similar type of computer vision to recognize the keyboard and relative hand position, although it is currently not the most accurate which is where our capacitively augmented system would be superior.

# Week of October 31st- November 6th

**Summary**

This week was the first time we had some hardware to test out. We had the Arduino Nano and MPR121 sensor boards in hand for initial testing. Along with that testing I researched communication protocols for serial busses utilized by the Arduino and devices which turned out to be the I2C protocol. although our priority is getting a working prototype, we also did some physical work with the keyboard like disassembling it and planning the placement of our components just to keep that in the back of the mind.

**Resources**

https://learn.sparkfun.com/tutorials/i2c/all

https://forum.arduino.cc/t/multiple-mpr121-capacitive-sensors-addresses-and-installation/453518

https://forum.seeedstudio.com/t/12-key-capacitive-i2c-touch-sensor-v2-mpr121-multiple-boards/252121

https://arduino.stackexchange.com/questions/17738/mpr121-multiple-using-breakout

https://files.seeedstudio.com/wiki/Grove-I2C_Touch_Sensor/res/Freescale_Semiconductor;MPR121QR2.pdf

# Week of November 7th- 13th

**Summary**

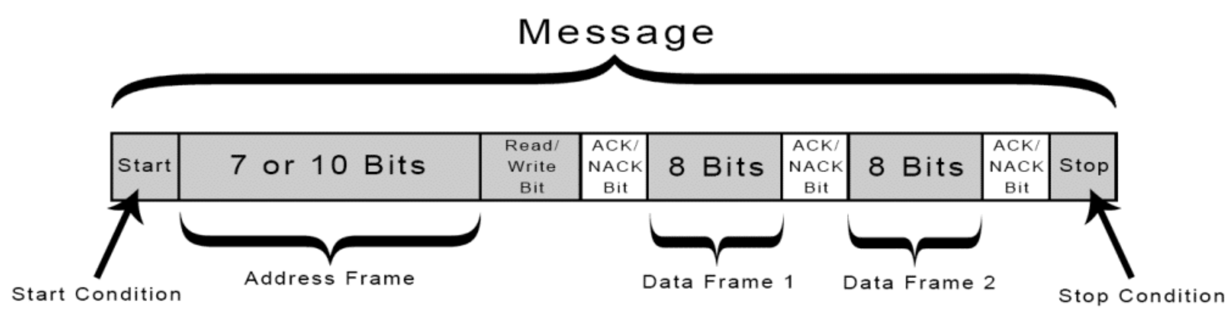
This week I was mainly focused on testing our Arduino/Sensor combo functionality. I got the necessary components soldered to the sensor breakout boards to test multiple devices over the I2C bus which is used by the sensors to communicate with the Arduino.

**Discussion**

The I2C protocol uses aspects from both Universal asynchronous receiver-transmitter (UART) and Serial Peripheral Interface (SPI). It uses only two wires to communicate Serial Data (SDA) and Serial Clock (SCL) which is nice. It works by having a master (the Arduino) and slave devices (MPR121's) which is nice for multiple devices on a single communication link and wire. The MPR121 boards allow four different addresses to be selected for differentiating between devices on the I2C connection. This means we can have up to four MPR121's connected to a single Arduino allowing for up to 48 individual capacitive touch inputs. This number was plenty up until we started doing some thinking about certain edge cases with our planned grid layout. The grid layout would theoretically work great with some data manipulation and thresholds in out software, until n-key rollover is desired or a lack of sensitivity from the hardware. Consider the simple case of all eight fingers on the home row. In this scenario the input connected to the home row would be activated at some capacitance, and the eight inputs connected to the corresponding column would be activated. If your first finger were to move from the J key to the u key then the single row sensor would need to have the capability to distinguish between the capacitance of eight fingers on that row and seven in order to properly display what keys were being touched. There might be other ways to get around this issue like implementing software which is smart enough to know a there should only be eight keys activated at a given time, but with accuracy and complexity in mind, we concluded that we should have a dedicated touch input per key for maximum accuracy and reliability. With the grid layout, connecting 61 keys only required up to 2 MPR121 boards, however, if each key is individually connected to a touch sensor input, we would require a full 61 inputs exceeding the 48 maximum per Arduino. We will now have to resort to the use of a multiplexer or similar device to have enough touch inputs with one Arduino.

**Resources**

| | |
|---|---|
| **Wires Used** | 2 |
| **Maximum Speed** | Standard mode= 100 kbps |
| | Fast mode= 400 kbps |
| | High speed mode= 3.4 Mbps |
| | Ultra fast mode= 5 Mbps |
| **Synchronous or Asynchronous?** | Synchronous |
| **Serial or Parallel?** | Serial |
| **Max # of Masters** | Unlimited |
| **Max # of Slaves** | 1008 |




https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/

https://www.google.com/search?q=I2C+MULTIPLEXER&rlz=1C1GCEA_enUS930US930&sxsrf=AOaemvKMsdVL65c6ljP009VaYtwS__cJpw%3A1636954223124&ei=b_CRYdv2BoCiqtsPlqWnsAI&oq=I2C+MULTIPLEXER&gs_lcp=Cgdnd3Mtd2l6EAMyBQgAEJECMgUIABCRAjIFCAAQkQIyBQgAEIAEMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQyBQgAEIAEOggIABCABBCxAzoOCC4QgAQQsQMQxwEQowI6EQguEIAEELEDEIMBEMcBEKMCOgQIABBDOgcIABCxAxBDOgsILhCABBDHARCvAToECAAQDToGCAAQDRAeSgQIQRgAUABYzh5gyB9oBXACeAGAAeQGiAGMGJIBDDEwLjYuMS4xLjYtMZgBAKABAcABAQ&sclient=gws-wiz&ved=0ahUKEwib6u3X0Zn0AhUAkWoFHZbSCSYQ4dUDCA4&uact=5

# Second Semester

# To February 13th

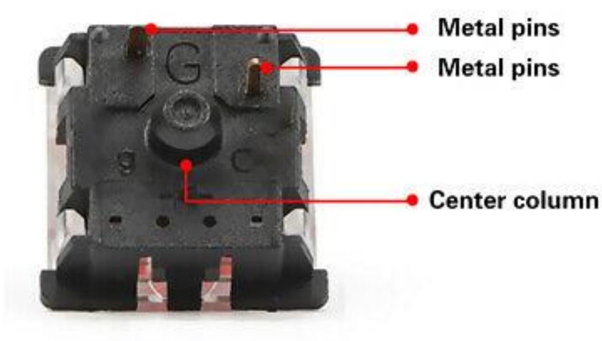
### Summary

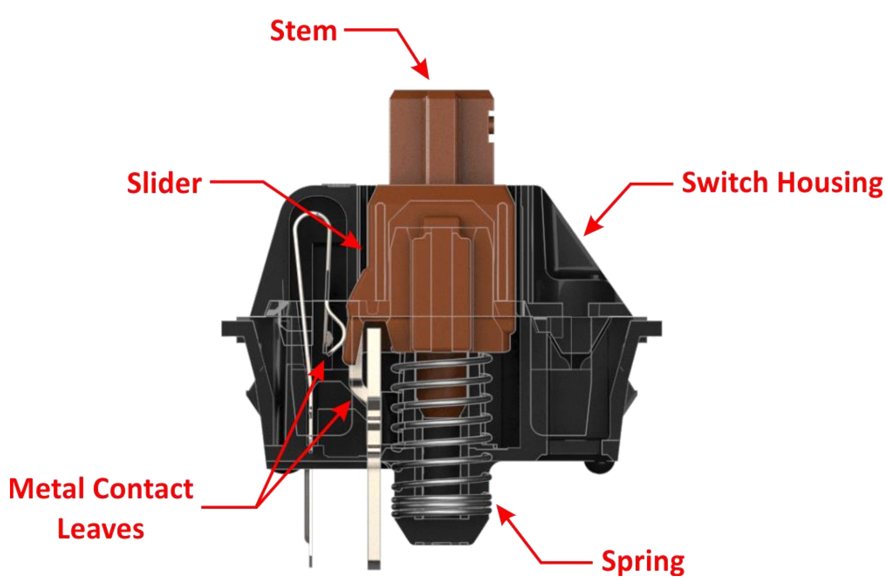
Beginning the semester with all the parts I had ordered at the end of last semester, I had quite a few items to setup before doing any testing. The testing priority for me was verifying the viability off copper tape placed under the keycap.

### Discussion

Last semester we ended with our prototype running with aluminum tape on top of keycaps to have direct contact with the conductive surface. Now that we had the correct copper tape which accepts soldering (aluminum oxidizes so fast it is nearly impossible to solder to) I could test if our sensor equipment was sensitive enough to have indirect touch sensing. After setting up everything I measured about a 10-count difference when indirectly sending compared to a 50-count delta with direct touch. This showed our sensors could work with thee right calibration.

We were also grappling with the problem of how and where to route and attach all the wires. We have come up with a few ideas to route a wire up through the actual key switch which would make a much cleaner routing of wires. In the picture below a hole could be drilled into the bottom of the center coloumn of the switch which a wire could be routed through. Then connecting the wire to the bottom of the stem (second picture) which would be coated in a conductive coating. This would allow the switch cap to be connected to the wire. Still a work in progress but so far it seems promising.


Metal pins
Metal pins
Center column


Stem
Slider
Switch Housing
Metal Contact
Leaves
Spring

# February 13th – 27th

**Summary**

Quite a few things were happening this period. First, I researched and ordered the keyboard we will use which has swappable switches. This is important for our plan to get a conductive path to the keycap. Second, I prepped the hardware for final testing before starting building including initial testing of the multiplexer connected with daisy chained sensors. Third, I researched and ordered a conductive coating spray we are planning to use. Fourth, I researched and ordered some of the last materials we will need to build our final product. Fifth, I started designing the layout of our hardware inside the keyboard after receiving it. Sixth, I designed the final hardware schematic. Seventh and finally I did some more thinking about the possible solution to making a conductive path from underneath the keyboard PCB to the keycap.

**Discussion**

1. The full-size keyboard we have been planning on using had one issue that started coming to light as we continued to develop our plan to route wiring to the keys. That issue being soldered PCB mounted switches. They keyboard I found has a hot-swappable PCB for easy removal and modification of the switches. In relation to this, I also ordered Gateron Brown switches which give tactile feedback when pressed and are a standard Cherry MX style.
2. The prepping of hardware was mostly straightforward. One notable decision was to only attach three sensors to two of the multiplexer ports. This will allow the sensors to be daisy-chained to maximize space efficiency and minimize wiring.
3. Our team settled on trying to use a conductive spray coating as the intermediary conductor from the wire to the keycap. I found the brand MG Chemicals who make four different options for conductive spray coatings. These sprays are also advertised to adhere well to most injection molded plastics which fits our application well. Carbon was immediately counted out for its higher surface conductivity compared to the other three. Next nickel was discarded from the running due to nickel being considered a human carcinogen. So, with the choice being between Silver-coated copper and just silver, the former was the clear option due to the significantly lower cost with similar

properties like conductivity. Below is a comparison chart.

## Conductive Spray Paints Comparison Chart

| | 838AR-340G | 841AR-340G | 843AR-340G | 842AR-140G |
|---|---|---|---|---|
| **UNCURED PROPERTIES** | | | | |
| Conductive filler | Carbon | Nickel | Silver-coated copper | Silver |
| Format | Aerosol | Aerosol | Aerosol | Aerosol |
| Color | Black | Dark grey | Light metallic brown | Light grey |
| Percent solids | 15% | 57% | 31% | 61% |
| Density @ 25 °C [77 °F] | 0.84 g/mL | 1.34 g/mL | 0.99 g/mL | 1.38 g/mL |
| Viscosity @ 25 °C [77 °F] | 46 cP | 61 cP | 87 cP | 80 cP |
| Calculated VOC | 587 g/L | 470 g/L | 404 g/L | 361 g/L |
| Theoretical coverage @ 2 mil (based on 100% transfer efficiency) | 20 016 $cm^2$ | 44 785 $cm^2$ | 23 290 $cm^2$ | 46 000 $cm^2$ |
| Recoat time | 3 min | 3 min | 3 min | 3 min |
| Cure time @ 22 °C [71.6 °F] | 24 h | 24 h | 24 h | 24 h |
| Cure time @ 65 °C [149 °F] | 30 min | 30 min | 30 min | 30 min |
| **CURED PROPERTIES** | | | | |
| Resistivity | 0.69 Ω·cm | 0.0076 Ω·cm | 0.00022 Ω·cm | 0.00012 Ω·cm |
| Surface resistance @ 50 µm | 110 Ω/sq | 0.60 Ω/sq | 0.080 Ω/sq | 0.050 Ω/sq |
| Salt fog resistance @ 35 °C [95 °F], 96 h | Excellent | Excellent | Poor | Excellent |
| Constant service temperature | -40–120 °C [-40–248 °F] | -40–120 °C [-40–248 °F] | -40–120 °C [-40–248 °F] | -40–120 °C [-40–248 °F] |
| Adhesion (ABS/PC) | 5B | 5B | 5B | 5B |
| Pencil hardness | H, hard | 3H, hard | F, medium | 3H, hard |
| Magnetic class | Diamagnetic | Ferromagnetic | Diamagnetic | Diamagnetic |

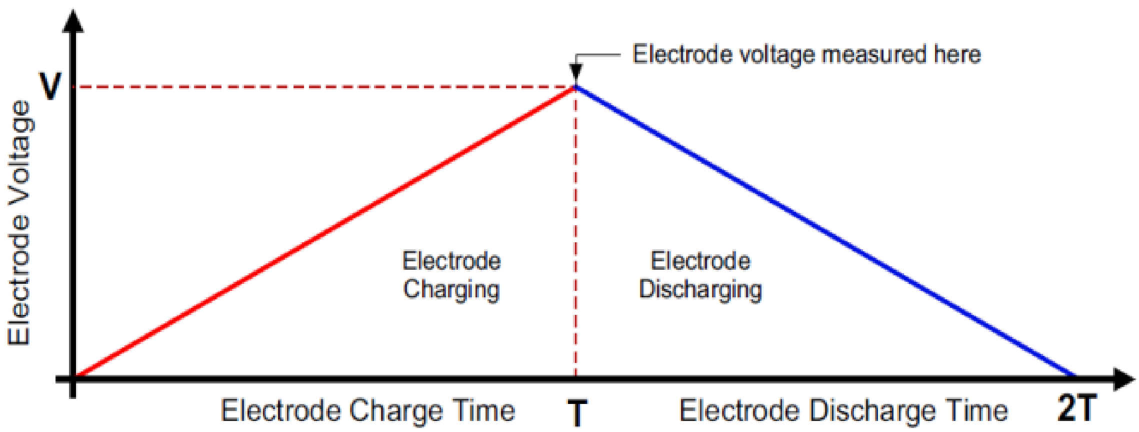
Silver-Coated Copper:

### 843AR-340G — Super Shield Silver-Coated Copper Conductive Paint

This copper conductive coating for plastic consists of a 1-part, solvent-based acrylic lacquer, pigmented with a highly conductive silver-coated copper flake. It is smooth, hard, and abrasion resistant. It is a ready-to-spray system, with no let down necessary. It has a quick dry time, with no heat cure necessary. It adheres strongly to most injection-molded plastics, such as ABS, PVC, nylon and polycarbonate.

Copper conductive paint is mostly used to provide a conductive coating for the interior of plastic enclosures to suppress EMI / RFI emissions. It excels when higher levels of shielding are required, giving near the performance of a silver filled coating at significant savings. It may also be used as a conductive undercoat for electroplating.

To show why a lower resistance is favorable I will use this illustration of how the capacitive touch sensor measured the change in capacitance to sense a touch. In short, the touch sensor applies a charge current I over a period of time T then measures the voltage V as shown in the graph. The lower the voltage, the lower the ADC count value and greater the difference between baseline and reading. The next picture shows the equation of how capacitance, charge, current, time, and voltage relate. Another important attribute of the MPR121 is that it uses a constant DC charge current scheme for capacitance measurement. With all this information resistance shouldn't affect the capacitance readings because the sensor should automatically adjust voltage to keep a constant charge current over the specified charge time. This only works to a certain point though because the constant current supply can theoretically only supply a certain maximum voltage. This is my current theory. Given this It is my opinion that we pick the lowest resistance conductive spray for efficient operation.

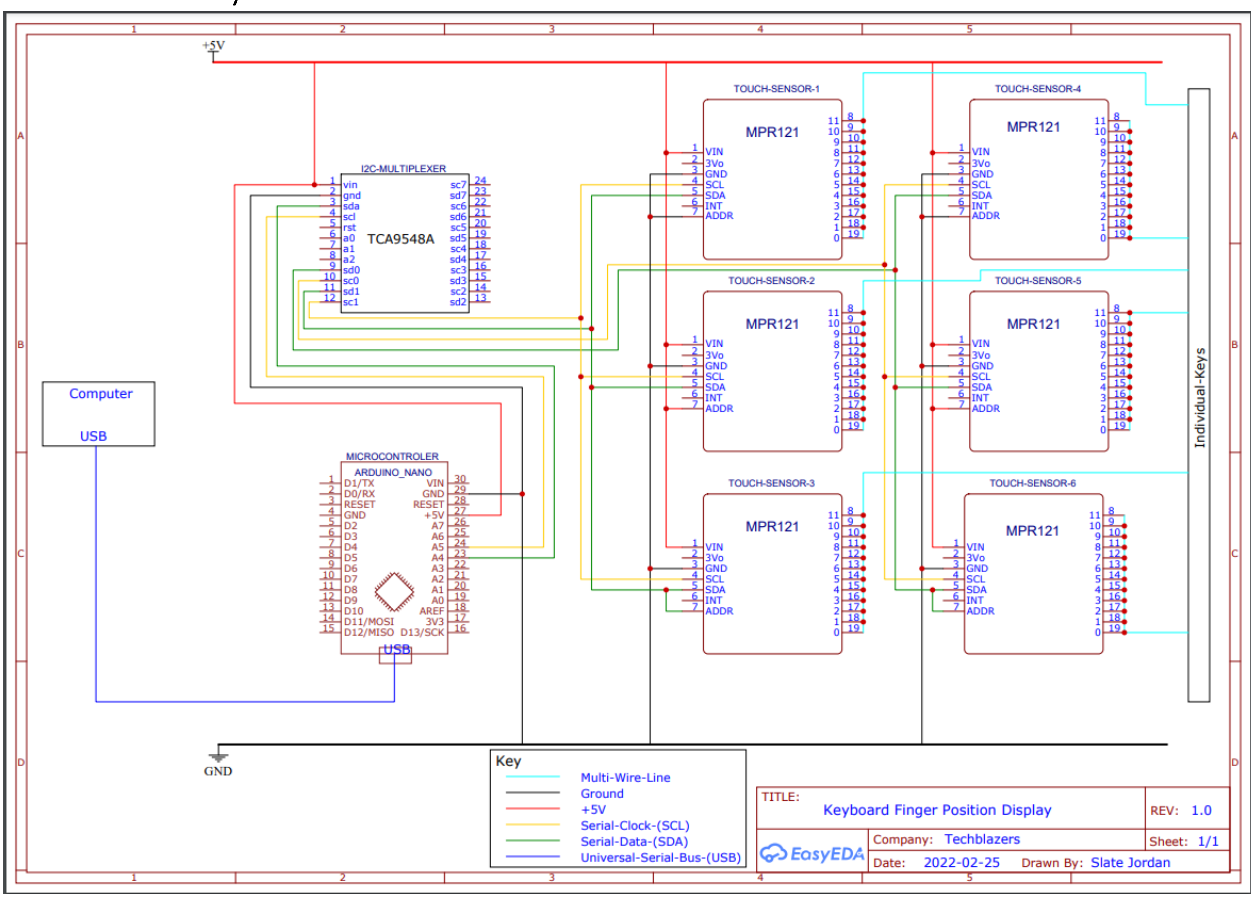



The amount of charge (**Q**) applied is programmable by setting the charge current (**I**), and the charge time (**T**). Once the electrode is charged, the peak voltage (**V**) at the end of charge is measured by internal 10bit ADC. This voltage V (that is the ADC counts) is reverse proportional to the capacitance (**C**) on the sensing channel.

$$C = \frac{Q}{V} = \frac{I \times T}{V}, \qquad V = \frac{Q}{C} = \frac{I \times T}{C}$$

4. Nothing much to cover on ordering the last materials. I found some lead free solder wick for disassembling pins on our Arduino and sensors for final assembly. Also found some multi color wire for when we assemble to provide easier troubleshooting rather than having 61 wires all the same color.
5. We are working with limited space inside the keyboard case since it is a 60 percent keyboard. With 8 pieces of hardware needing to be fit, I looked at some potential layouts inside the case keeping in mind the Arduino's USB port needing to have an access hole cut on the back. The case has some support fins which I am planning on taking out to provide space for sensors (see picture). Since the case has a tilt, the rear will have more space so that is where the hardware will reside. see picture for rough layout. As I mentioned earlier two groups of three daisy-chained sensors will allow for less wiring.


6. The final hardware schematic shown below shows how the hardware will be wired together. The connections to individual keys are arbitrary because software will accommodate any connection scheme.




7. The keys arrived the the 27th so I haven't had much time to experiment, but it has given me a more concrete idea of how passing the wire throuhg the stem might be possible. The spring could also be utilized as a conductor as another option. The wire could be soldered to the spring and spring in contact with conductive coated slider.