CS492-PUI Term Project Final Report

Team number	4
Project name	GetSetPress
Project title	Enhancing typing performance of touch keyboards with activation delay for neuromechanical control.
Team members	Auejin Ham
Proposal date	6/17/2021

PUI Problem to Address

Performance Differs by Button Modalities

It is an old question first proposed to HCI society in the 90s.

Research Studies	Touch (WPM)	Mechanical (WPM)	Targeting
Gould et al. (1990)	12	**20**	Novice Users
Seras (1991)	25	**58.4**	Experts
…	…	…	…
Grubert et al. (2018)	11.6	**26.3**	VR Typing Interface

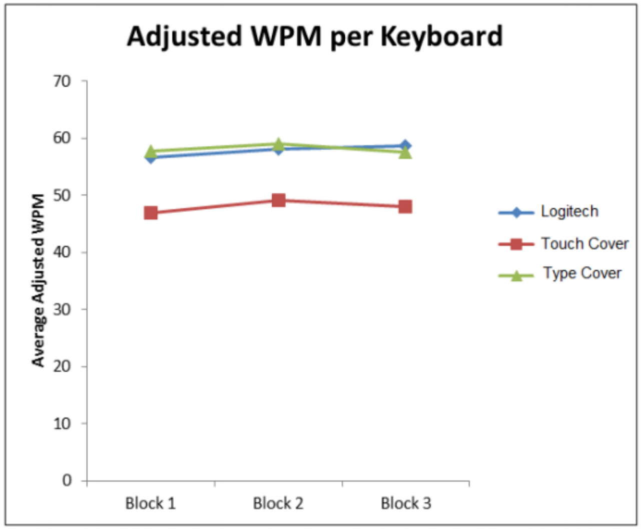
… but by what reason?

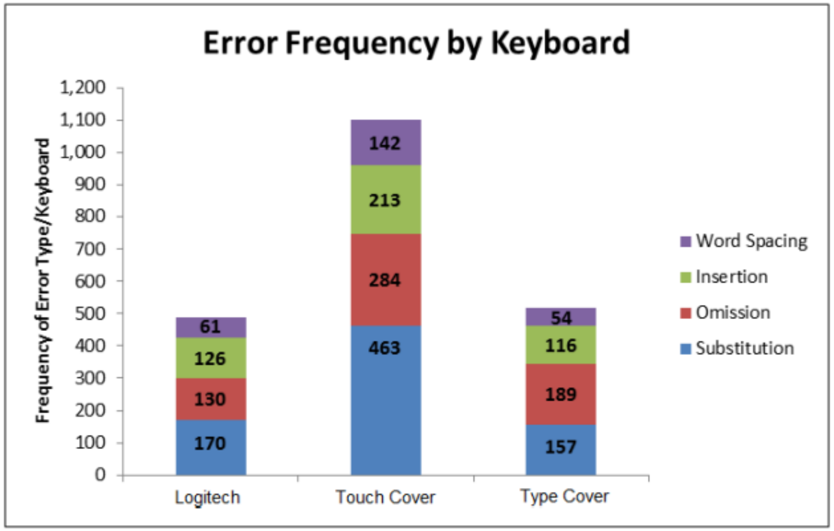
Performance in Details

Chaparro et al. (2014) compared different keyboards of Microsoft Surface.

Higher WPM of **mechanical** keyboards than the touch-based one.

Largest amount and frequency in **substitution** errors compared to others.





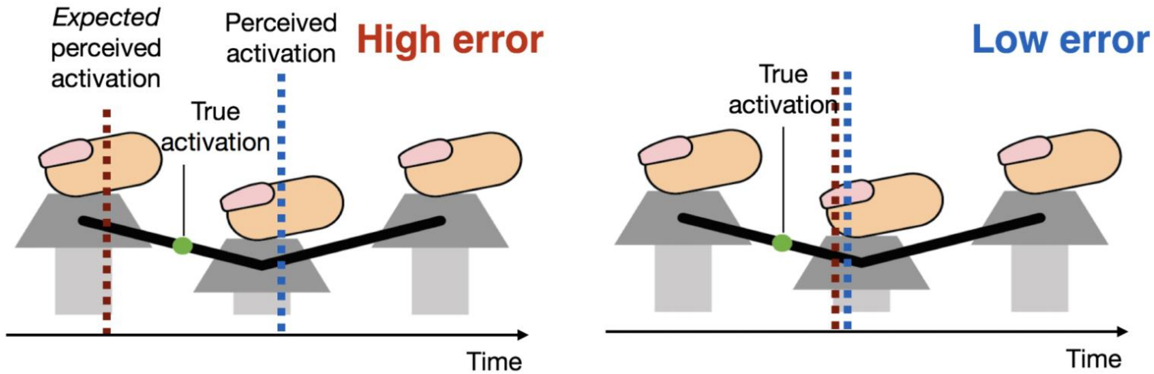
Existing Solutions

Substitution Error is the "Impostor"?

Why the substitution errors are so high for touch buttons?

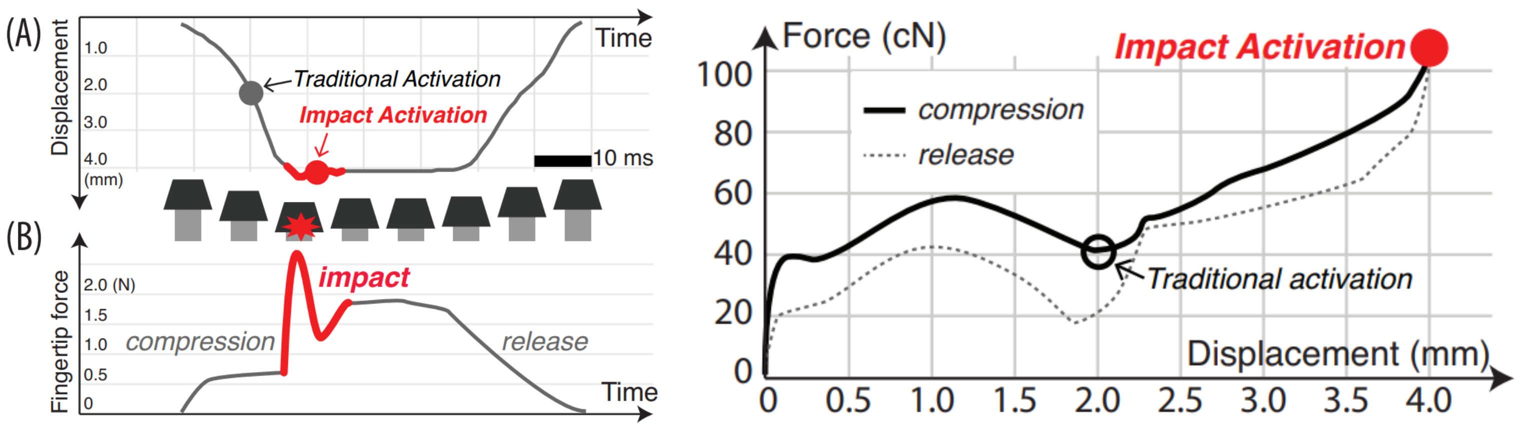
Neuromechanics of Button Press (Oulasvirta et al., 2018)

- button press as an open-loop problem based on cue integration theory
- the source of its mechanoreception simplified into a pulp compression, which correspond to indentation and retraction of the fingertip
- "To account for known aspects of temporal control better, models of sensory synchronization should be considered."



When to Confirm the Key Press?

Impact Activation (Kim et al., 2018)

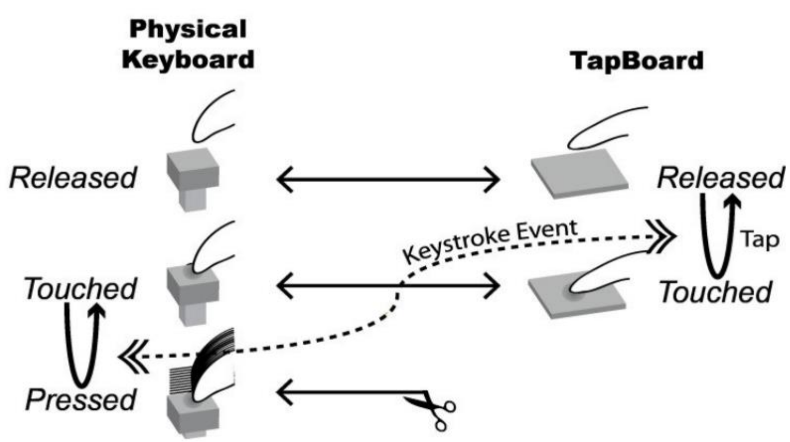


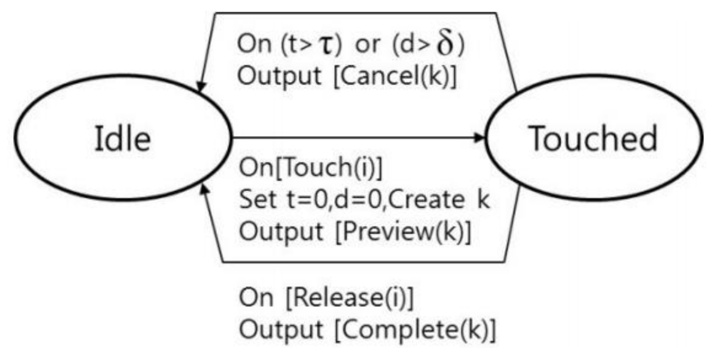
- superior to the traditional button activation methods for all performance metrics in a fast tapping task: higher success rates, less asynchrony, and less variance.
- performance gains (success rate and synchronicity) were greater in the physical button condition than in the touch button condition.
- performance gains with multiple buttons are not studied.

TabBoard (Kim et al., 2013)

mapping the states of a physical (mechanical) key to those of a touch button, enabling fingers to rest on the surface just as physical keyboard

no comparison of typing performance with mechanical keyboard, but notable since temporal threshold is applied for each key to be activated





Touch-Sensitive Mechanical Keyboards

PreSense (Rekimoto et al., 2003)
capacitance sensing with copper tapes and multiplexers, 20 keys

Dgts (Habib et al., 2009)
CellMatrix pattern working in loading mode, 96 sensors of AD7147 to detect finger blob

TabBoard (Kim et al., 2013)
capacitance sensing with copper tapes and multiplexers, 48 keys

FlickBoard (Tung et al., 2015)
CDC design of SmartSkin (shunt mode), measuring the change of mutual capacities at the sensor grid intersections

Impact Activation (Kim et al., 2018)
Hall-effect sensor (SS49E) for the depth of a mechanical button, and capacitance sensor (Adafruit MPR121) for a touch button, 1 key

Touch Sensitive Keyboards for VR (Otte et al., 2019)
capacitance sensing, no multiplexers thanks to Arduino Mega 2560 Rev3, 62 keys

PalmBoard (Yi et al., 2020)
utilized a commercially available pressure-sensitive touchpad (Sensel Morph)

Our Solution

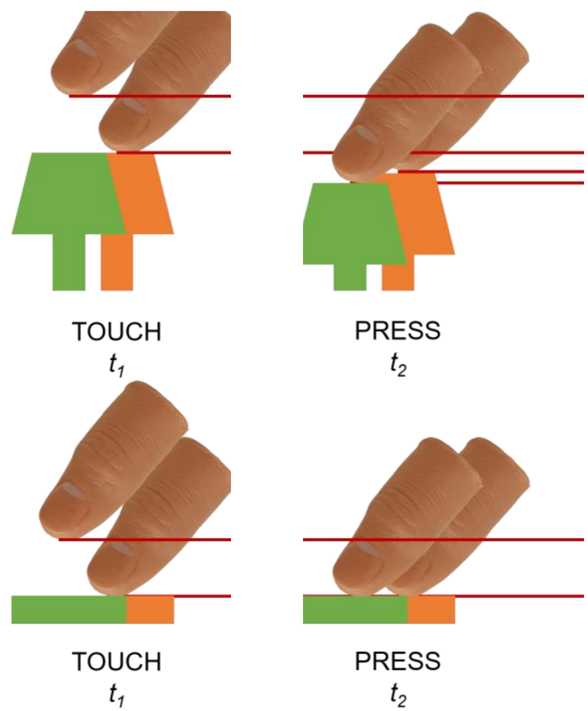
Our Solution

Neuromechanics of button press can be thought as
an open loop control problem (Oulasvirta et al., 2018).

Touch keys have no gap from touch until pressed,
which results no time to calibrate force of each different fingers.

Touching Order

Orange → Green



Pressing Order

Green → Orange

Orange → Green

Our Solution

Our Prototyping Goal
- touch keyboards with similar performance of physical keyboards

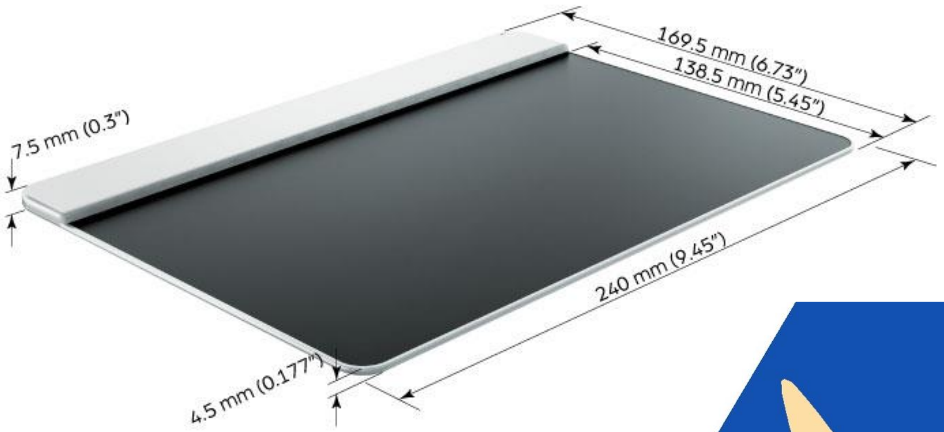
1. We want to reduce substitution errors of touch-based keyboards.

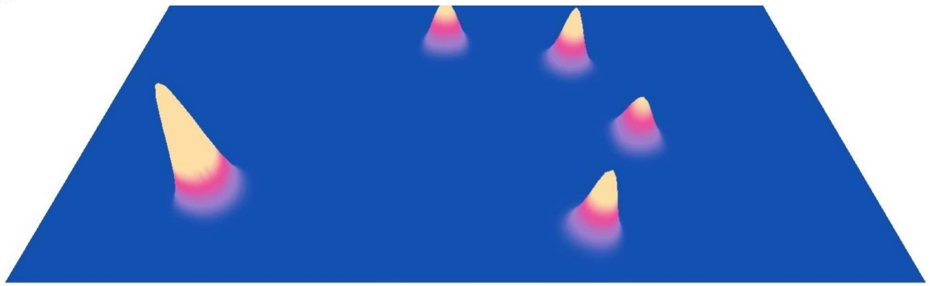
1. If people calibrate the order of buttons from touch until pressed, offering a spatiotemporal margin to do so will reduce the error.

1. We will produce the margin based on impact activation as an initial value, which later tuned as the time stabilizing press order of touched keys.

Apparatus: Touch Keyboard

Hardware Specification





Sensel Morph: an array of highly-sensitive pressure sensors

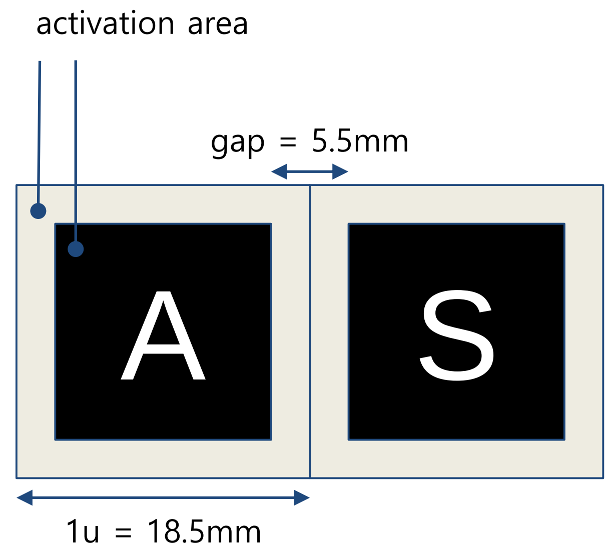
- 185 x 105 sensor elements at a 1.25mm pitch
- 5g - 5kg sensing range per touch
- Each contact can sense approximately 30,000 levels
- High Speed Mode: 500 Hz (2 ms latency)
- 6502 dpi

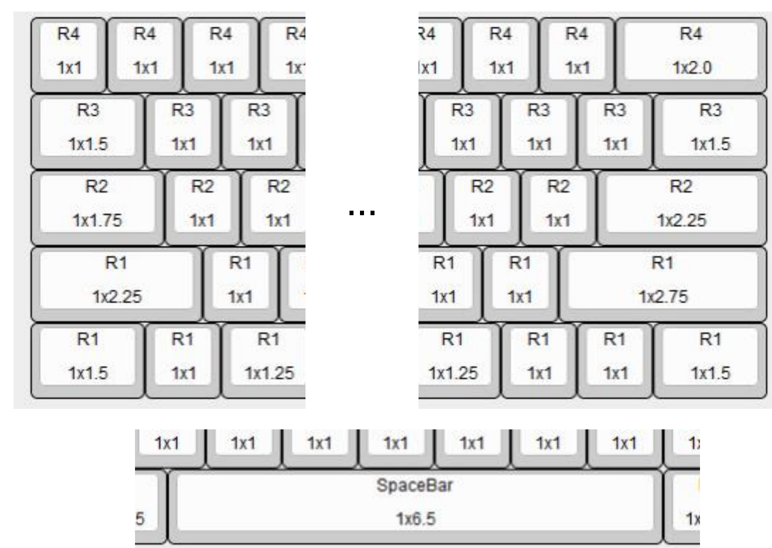
Keyboard Layout

Standard 1u keycap of Wooting keyboard
- 1u: 18.5 mm, margin: 2.75mm

Layout of 50 keys from Logitech G710
- its dimension minimized into 13u X 5u

Removable stickers attached for layout stability





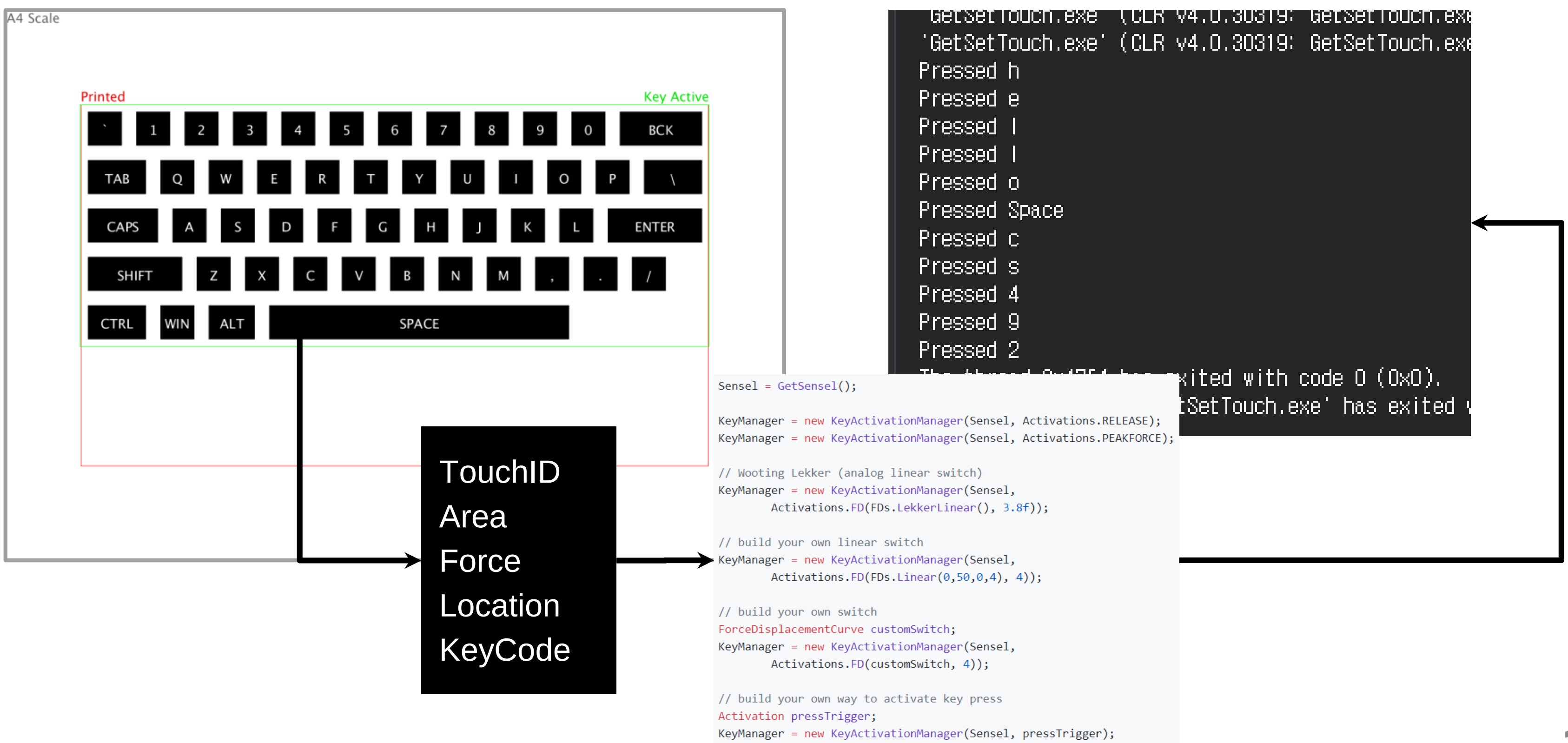
Customizing Activation Points

```
Sensel = GetSensel();

KeyManager = new KeyActivationManager(Sensel, Activations.RELEASE);
KeyManager = new KeyActivationManager(Sensel, Activations.PEAKFORCE);

// Wooting Lekker (analog linear switch)
KeyManager = new KeyActivationManager(Sensel,
        Activations.FD(FDs.LekkerLinear(), 3.8f));

// build your own linear switch
KeyManager = new KeyActivationManager(Sensel,
        Activations.FD(FDs.Linear(0,50,0,4), 4));

// build your own switch
ForceDisplacementCurve customSwitch;
KeyManager = new KeyActivationManager(Sensel,
        Activations.FD(customSwitch, 4));

// build your own way to activate key press
Activation pressTrigger;
KeyManager = new KeyActivationManager(Sensel, pressTrigger);
```

Event Handler Design

Different activation methods available for each key

- TOUCH, RELEASE, custom FD design
- PEAKAREA, PEAKFORCE from Impact Activation (Kim et al., 2018)



Apparatus: Analog Keyboard

Hardware Specification

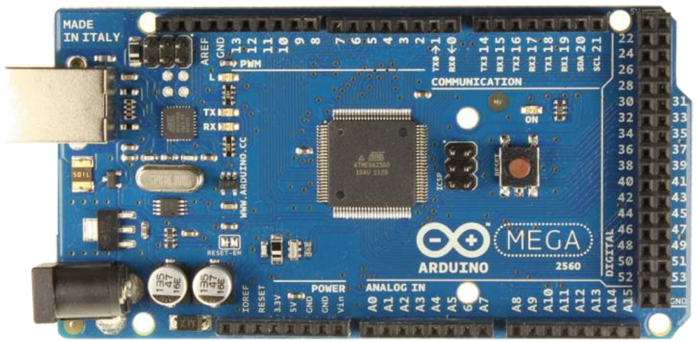
Switch : Flaretech Linear55 Red (analog linear)

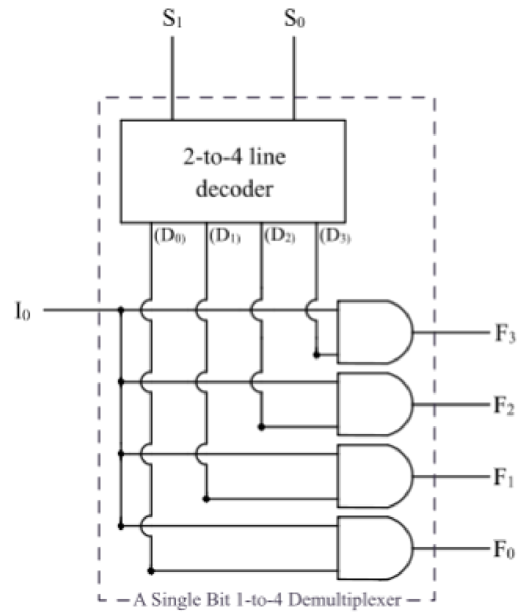
- key end force: 55cN
- travel distance: 4mm
- analog actuation range: 1.5 - 3.6mm

Wooting Keyboard One

- scan per key <1ms: (MUX to MCU)
- polling rate: 1000Hz
- matrix layout scan: 15-22ms

Wiring Keycaps



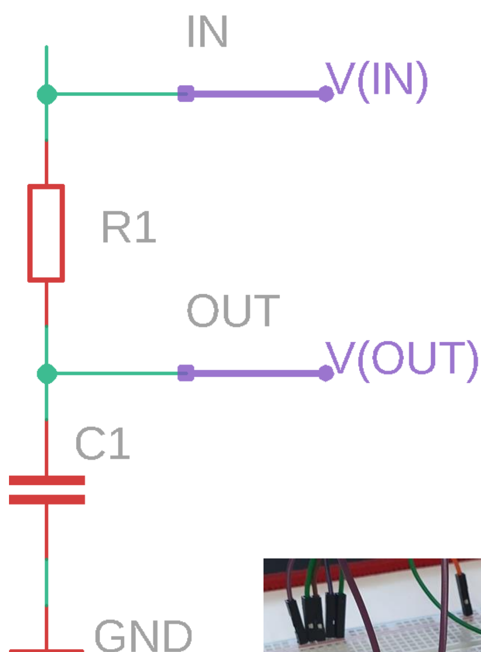


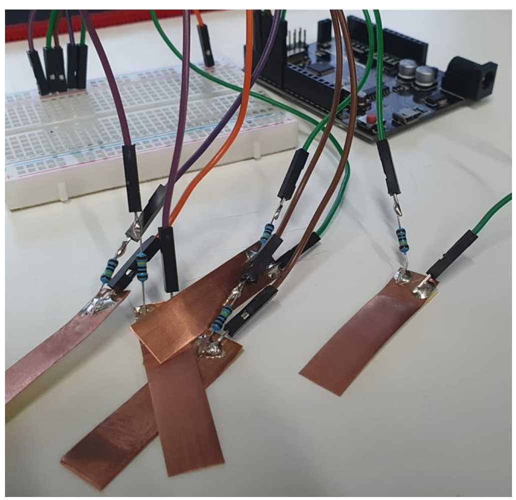
S_1	S_0	I_0	F_3	F_2	F_1	F_0
0	0	0	0	0	0	0
		1	0	0	0	1
0	1	0	0	0	0	0
		1	0	0	1	0
1	0	0	0	0	0	0
		1	0	1	0	0
1	1	0	0	0	0	0
		1	1	0	0	0

- Wooting keyboard (255 levels of press depth for each linear switch)
- Build capacitive keycaps with copper tapes
- Nylon-coated wires for stability
- Multiplexing every keycaps into a microcontroller.

All of these to collect stream of Touched, Pressed, and Typed Keys

Touch Detection via Capacitive Sensing?





Arduino STL <CapacitiveSensor.h> generates high frequency square wave and measures temporal delay of HIGH between IN and OUT

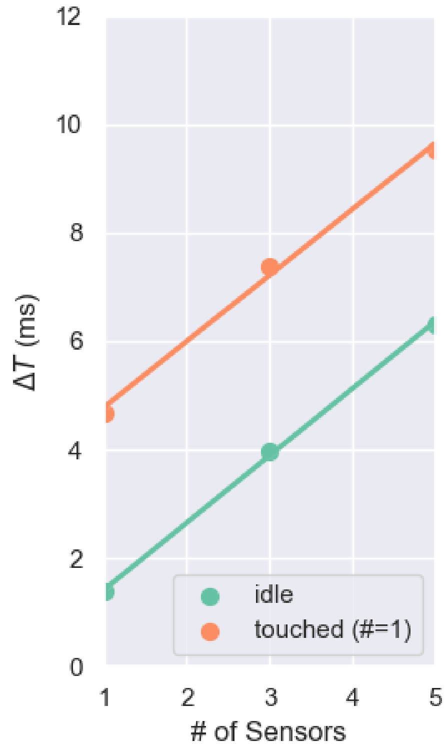
We soldered five sensors and measured following metrics
    delay to read every sensor
    size of V(OUT)

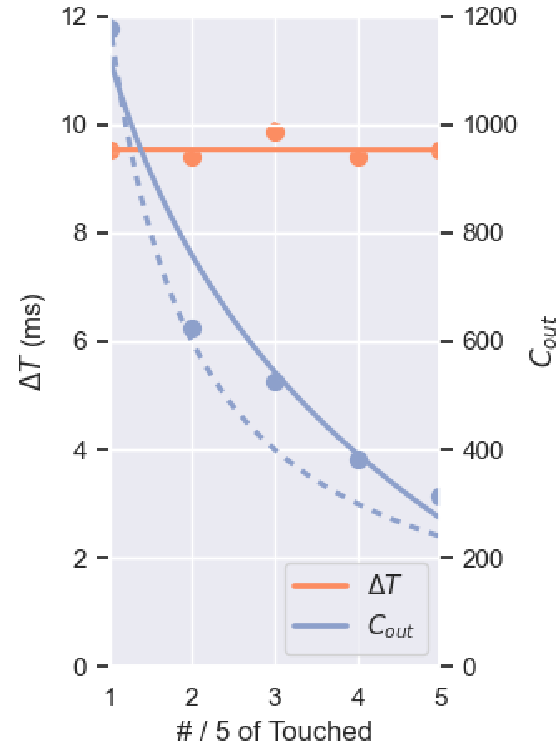
in different levels of
    # of connected sensors
    # of touched sensors

under following environment
    baud=115200
    chunk=140 (rows per print)

Too Slow for Wooting!

Delay must be **< 200 us**
to read 50 keys under 1ms
(polling rate of 1000 Hz).





of Sensors

- linearly increases delay
- extra const. delay when pressed

of Touched Sensors out of 5

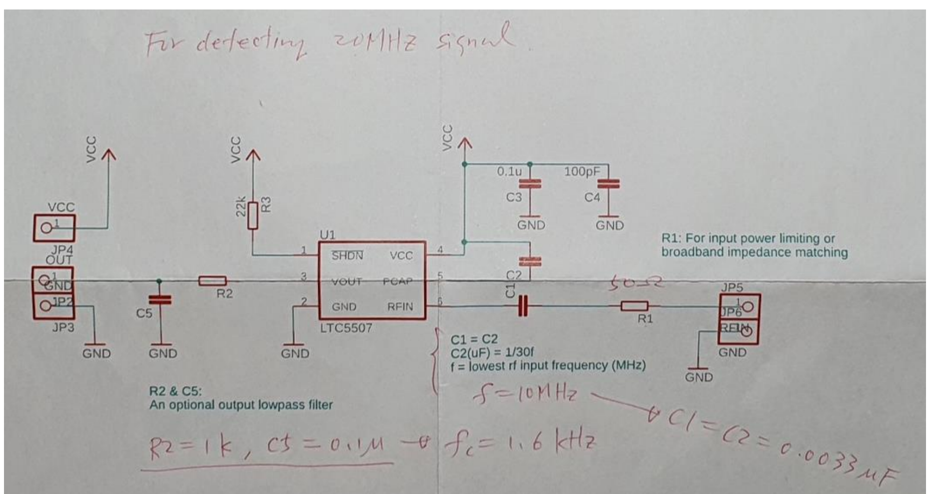
- does not change delay
- but change output threshold

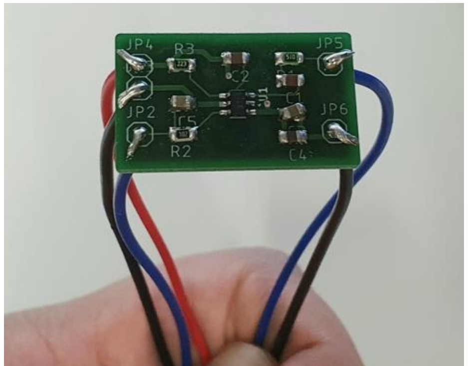
Touch Detection via Power Detector

We utilized Power Detector (LTC5507) to minimize detection latency.

Power detector returns DC Vin if specific amount of frequency were detected. Detection Frequency is tuned by capacities and resistors.

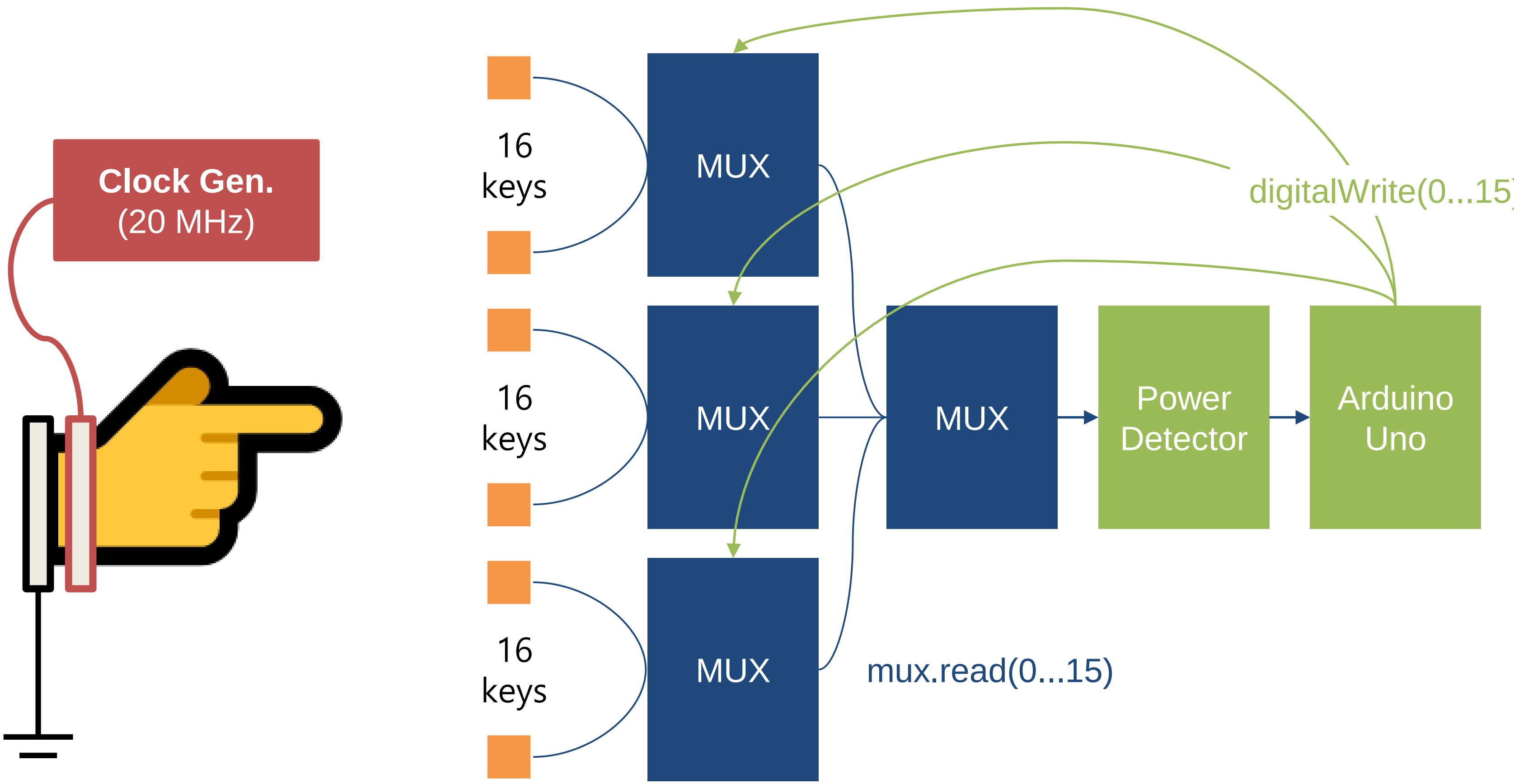
We tuned our power detector for 20 MHz square wave.



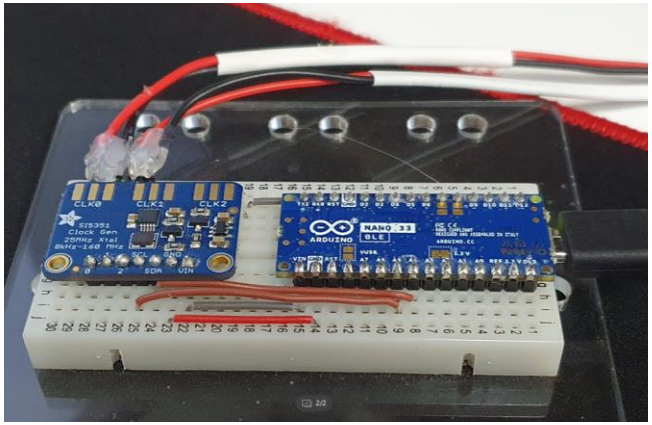


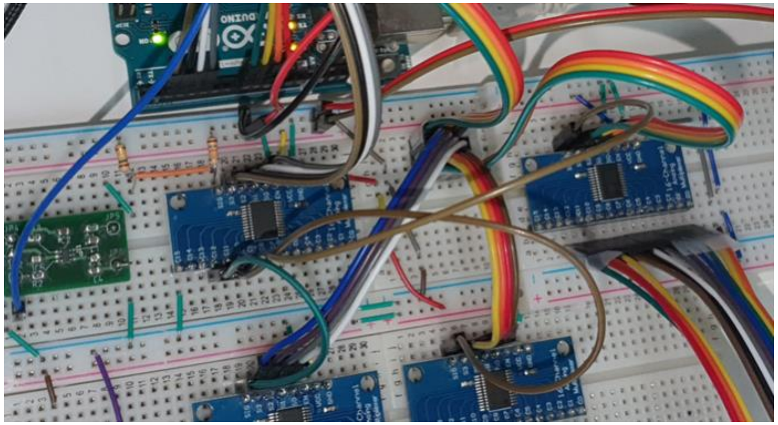
Schematic of Our PUI

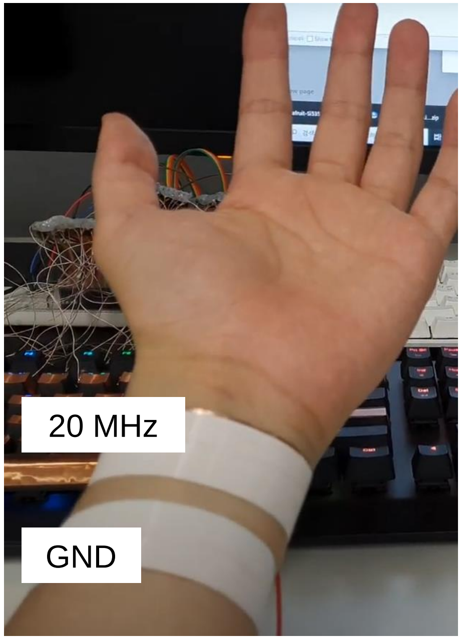
Clock Generator (SI5351)
Power Detector (LTC5507)
16-Pin Multiplexer (CD74HC4067) * 4

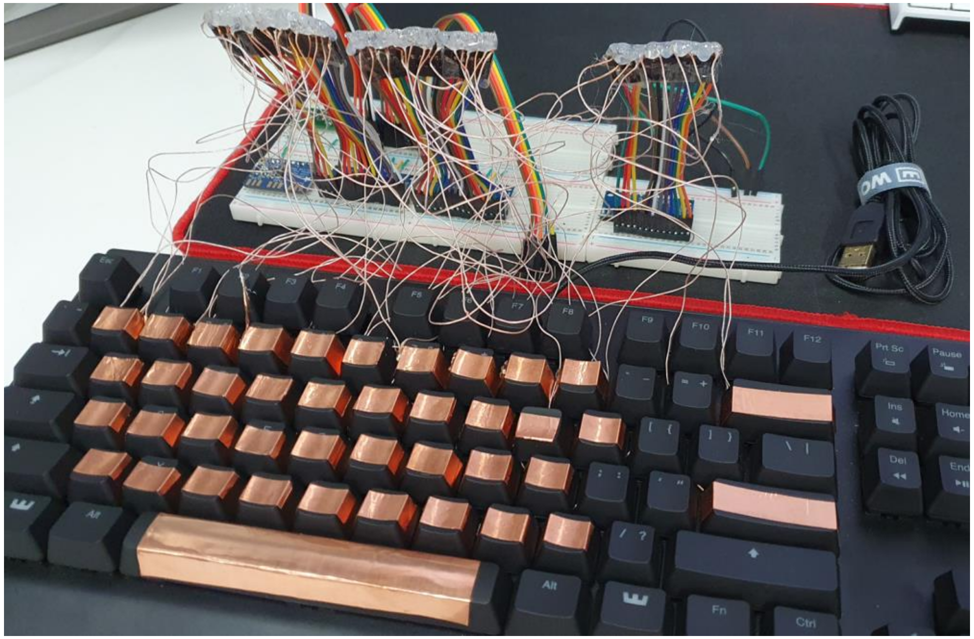


Analog Keyboard Touch Detector

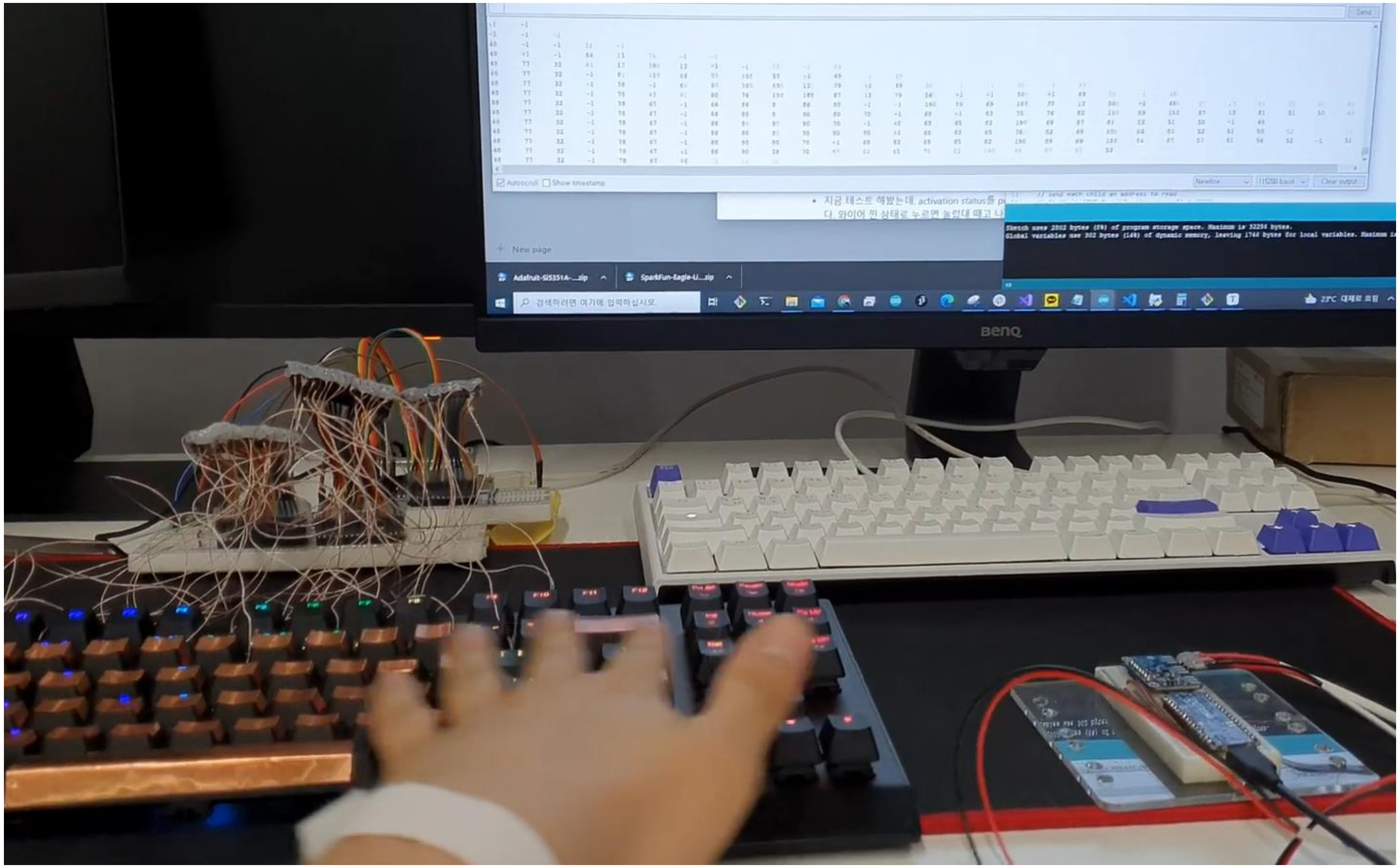








Touch Detection Performance Demo



https://youtu.be/0TLU3MaX__Q

Validating the Key-Logging System

Clock drift of touched, pressed, and typed order of keys must be measured and calibrated if recorded by different interfaces.

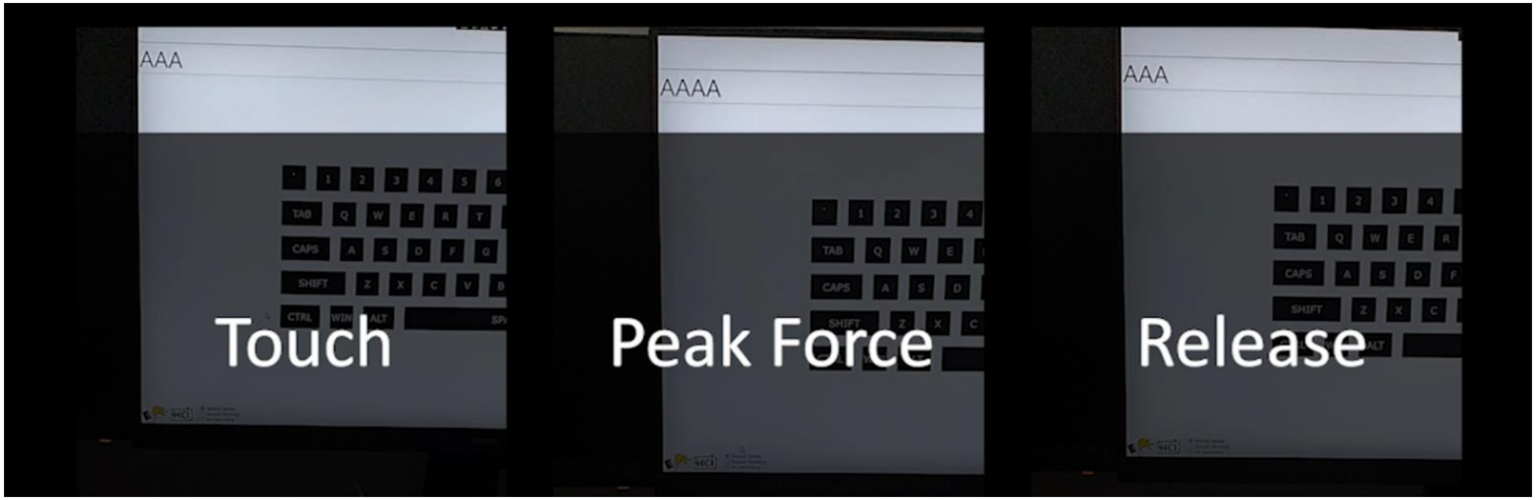
Even the order of pressed and typed keys might differ due to the limited bandwidth of a keyboard.

Validation Process

- check the bandwidth of USB communication of a keyboard
- obtain the time duration for each stage of touch-press-type

1. connect an LED to a capacitive keycap
2. create a fullscreen GUI that changes color for every keyboard input
3. record the LED and the GUI as a high frame-rate video
4. press keys several times and obtain the distribution of time duration

Validation - Touch Keyboard



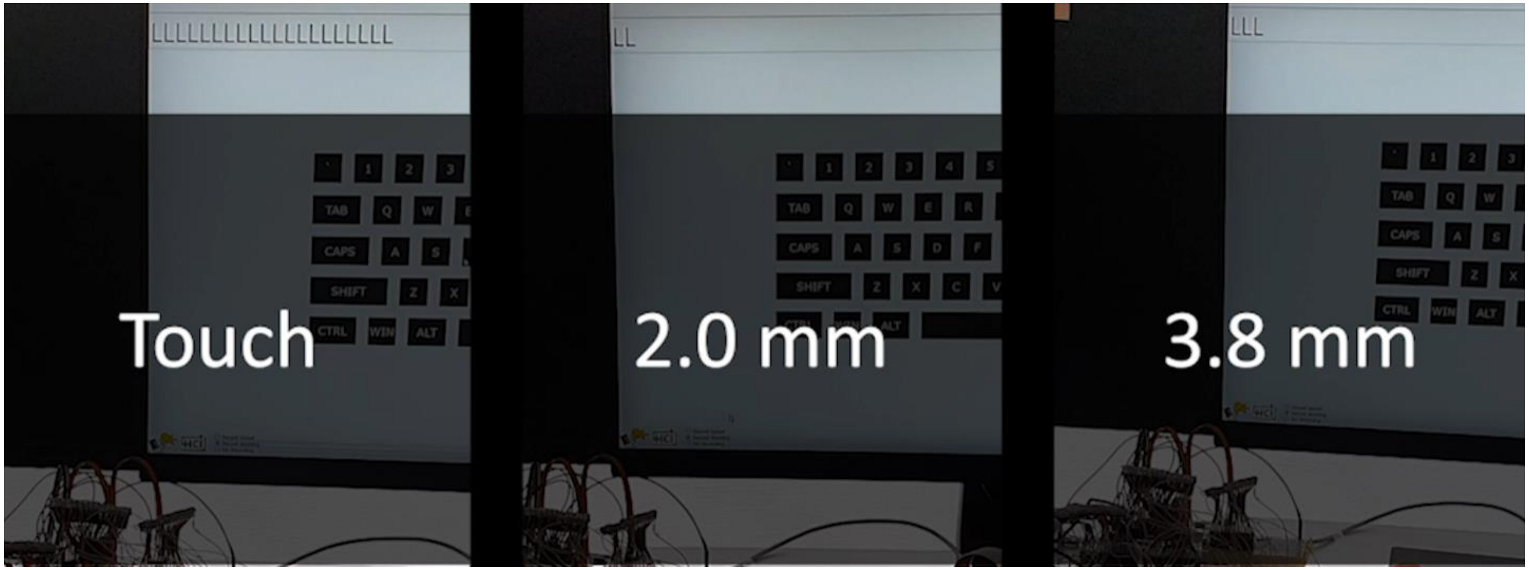
ΔT = 80.21 ms (SD=1.473)

ΔT = 139.9 ms (SD=50.71)

ΔT = 147.9 ms (SD=10.42)

https://youtu.be/rEIW__I0DTQ

Validation - Analog Keyboard



ΔT = 146.4 ms
(SD=6.771)

ΔT = 154.7 ms
(SD=43.23)

ΔT = 156.7 ms
(SD=12.50)

https://youtu.be/AtbNOHkr8Vk

Evaluation Plan

Conjectures to Validate

While using mechanical or touch keyboards, there will be difference in the order of touched keys and pressed (activated) keys.

Neighboring keys of given string sequence, and therefore order of fingers to be pressed, will affect the possibility of transposition error.

Activation point will affect the typing performance. Sufficient amount of delay on the activation will increase the typing performance.

Conventional Measures of Typing

Entry Rate (WPM, 5 chars including blank space as a word)
- conventional definition of word in typing interface

Character Error Distribution (# of insert, omit, transposition, etc.)
- conventional metric of Gentner et al. (1983)

Keystrokes per Character (error rate that includes editing effort)
- first introduced in Arif et al. (2009)
- later used in WiseType (Alharbi et al., 2019)

Qualitative Study
- interview session w/ likert scale measures
- perceived speed, accuracy, fatigue, overall satisfaction

Our User Experiment

40 phrases from the T-40 phrase set in Yi et al. (CHI 2018)
Participants instructed to type *"as quickly and as accurately as possible"*
If found error, participants can delete and retype it, or leave it uncorrected
1 minute break enforced between total 5 blocks

Block-wise measure of typing metrics for different activation points

1. Error Rate (CER), Typing Speed (WPM)
2. Ratio of Transposition Errors
3. Sequence Difference from Tactile and Activation Points

Activation Points

- Wooting.Activation = {Touch, 2.0*, MaxDepth}
- Sensel.Activation = {Touch, PeakForce**, Release}

* 2.0 = Travel Distance of Cherry MX Switch
** Impact Activation (Kim et al., 2018)

Observing Typing Log

```
YOU WOULDNT NOT BELIEVE YOUR EYES TE NILLION FLOES
OOOOUUUUUUU      QQQQOOOOOOOOOOOUUUUUUSSSSNNNNNTTTTTT    NNNNNNOOTTTTTTEEEEEEELIIIIEEEEEEEEEEEEEE     YYYYYYYUUUUUUUURRRRRRRRR     EEEEEEEEEYYYYYYEEESSSS     TTTTTTTEEEEEMMM    NNNNNIIIIILLLLLLLLII
IIOOOOOOONNNNNNN FFFFLLLLLLOOOOOOEEEEE
```

Logs of pressed and touched keys with depth of 3.8 mm for Wooting One.

Order of touch and pressed keys seems to be the same for these two logs. However, temporal distribution for each key were differed a lot. Maybe due to resting fingers?

With multiple blocks of experiments, we expect this could result typo.

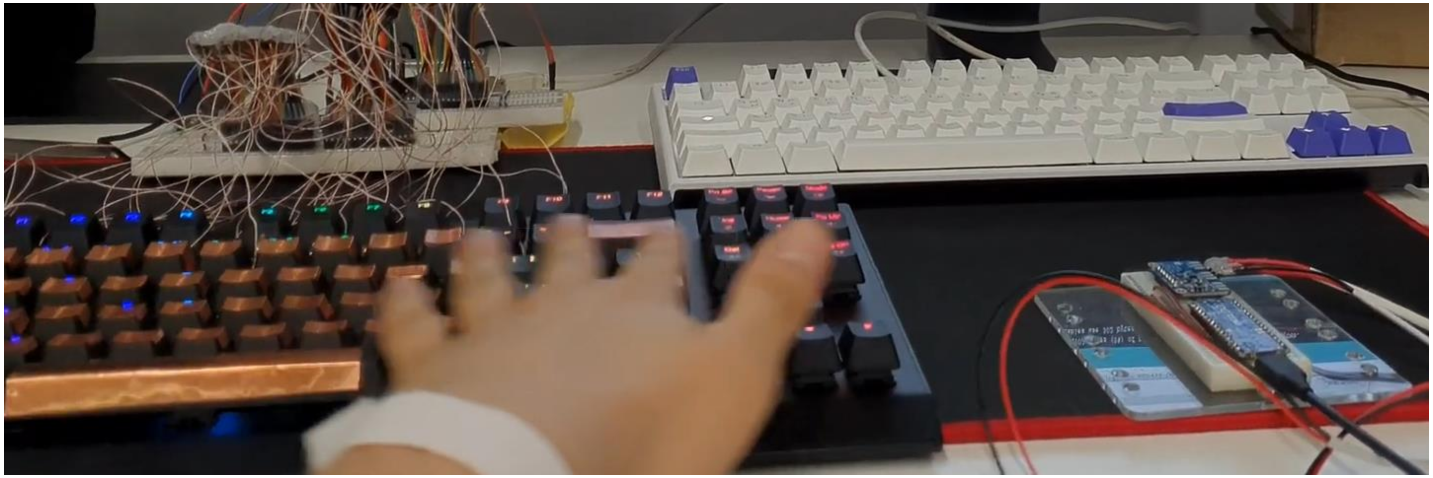
```
UYOU WOLDNTBEIEV URSMF
```

Logs of pressed and touched keys with simulated depth of 3.8 mm for Sensel Morph.
Depth simulated by FD curve of Lekker switches of Wooting keyboards.

Even under the same force threshold to activate, much higher error rates observed. This seems due to no tactile feedback during each key press.

Furthermore, applied depth showed less increment compared to Wooting logs (hammering fingers). This is due to different mental model of using touch keyboards.

Take-away Messages



1. Implemented a touch keyboard and a touch-sensing analog keyboard to test out the typing behavior of different (programmable) activation conditions.

2. Measured end-to-end latency and showed decent performance.

Tentative Plan by the Week

Week		Project Activity
9	4/26 ~	prototyping
10	5/3 ~	prototyping
11	5/10 ~	prototyping + experiment design
12	5/17 ~	feedback session
13	5/24 ~	prototyping + experiment design
14	5/31 ~	user experiment + data analysis
15	6/7 ~	data analysis + writing
16	6/14 ~	final presentation