

# Projekt-Vorschlag

Projekt: *PinkyPi*

Dokument: *Projekt-Vorschlag PinkyPi SS2024*

Version: *1.0.0*

Datum: *15.04.2024*

Status: *vorläufig final*

## Zusammenfassung

Unsere heutigen Tastaturen existieren mittlerweile seit knapp 150 Jahren in nahezu unveränderter Form und das trotz der sich stetig weiterentwickelnden Arbeitsweisen und Bedürfnissen. Ein ergonomisches Split-Keyboard bietet die Möglichkeit, diesen Fortschritt anzuerkennen und den Bedürfnissen moderner Nutzer gerecht zu werden. Durch die Aufteilung der Tastatur in zwei Hälften kann die Handhaltung optimiert und damit eine natürlichere Armposition ermöglicht werden, um Erkrankungen wie RSI (Repetitive strain injury) effektiv vorzubeugen.

## 1 Ziel

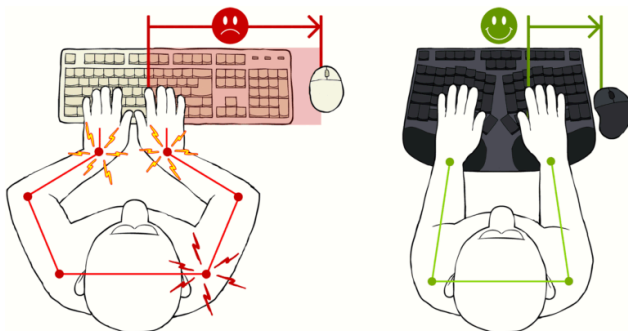
Das Hauptziel des Projektes besteht zunächst darin, einen funktionsfähigen Prototypen eines ergonomischen Split-Keyboards zu bauen, der eine Verbesserung zum Status Quo bezüglich Ergonomie darstellen sollte.

Darüber hinaus verspreche ich persönlich mir vom Projekt das Erlernen der PCB Design Software KiCad, um auch in Zukunft eigene Platinen für Projekte entwerfen zu können.

## 2 Ist-Situation

Wie bereits erwähnt, stammen unsere heutigen Tastaturen aus der Zeit der Schreibmaschinen, bei denen die Platzierung der Buchstaben so gewählt wurde, dass sich die Typenhebel nicht ineinander verfangen. Auch wenn diese Limitierung heutzutage keine Rolle mehr spielt, ist die traditionelle Tastenanordnung noch weit verbreitet.

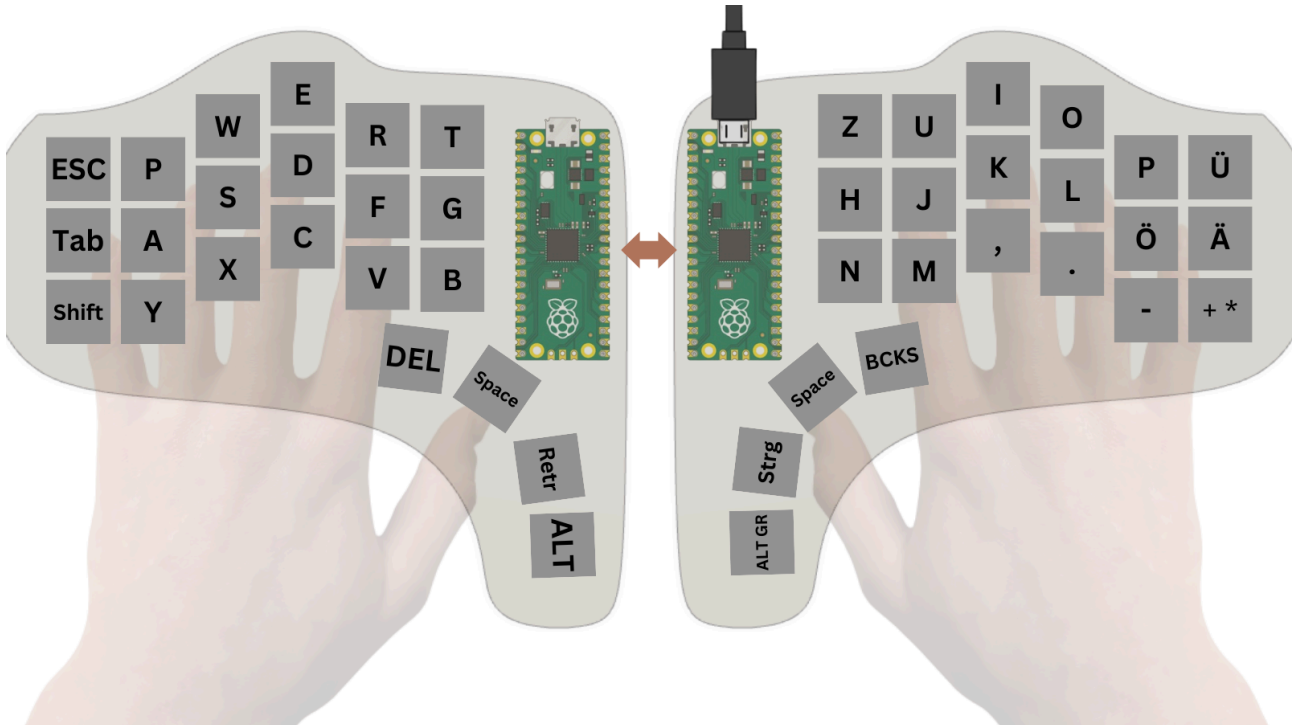
Da das Erlernen eines neuen Layouts jedoch viel Übung erfordert, werde ich mich in diesem Projekt lediglich auf die Entwicklung einer verbesserten Ergonomie konzentrieren, bei Bedarf lässt sich das Layout zudem jederzeit per Software verändern. Das größte Problem an herkömmlichen Tastaturen ist dabei das starke Abknicken der Handgelenke, wie die folgende Grafik veranschaulicht:



<https://trulyergonomic.com/ergonomic-keyboards/wp-content/uploads/Truly-Ergonomic-Cleave-Reduce-Conventional-Keyboard-Pain-27-768x403.gif>

### 3 Projekt-Ansatz

Das PinkyPi Keyboard besteht aus zwei Hälften, die jeweils aus 22 Tasten und einem Raspberry Pi Pico bestehen. Die beiden Hälften sind untereinander verbunden, wobei eine der beiden Hälften zusätzlich mit dem PC verbunden wird.



### 4 Deliverables

Wie auch schon im letzten Semester, wird zum PinkyPi Projekt ebenfalls eine Entwickler- sowie Benutzerdoku geschrieben, damit das Projekt genutzt und gewartet werden kann. Zusätzlich ist geplant, den Source Code und die Design-Dateien öffentlich auf GitHub zur Verfügung zu stellen, damit theoretisch auch andere an meinem Projekt weiterarbeiten können bzw. ich selbst jederzeit auf die Dateien zurückgreifen kann.

Idealerweise sollte die fertige Tastatur in den Räumlichkeiten der FH zur Show ausliegen, damit sie von anderen Studierenden und Besuchern (zum Beispiel am Tag der offenen Tür) ausprobiert werden kann.

## 5 Material

Die für das Projekt benötigte Hardware ist in der folgenden Tabelle aufgelistet:

### Stückliste

Posten	Bezeichnung	Einzelpreis	Menge	Gesamtpreis	Bezugsquelle
1.	RasPi Pico	-	2	10,99€	Amazon
2.	Gateron Ks-33 Low-Profile-Tasten schalter	-	44 (65er Packung)	35,99€	Amazon
3.	Keycaps	-	44 (volles Set enthält mehr Tasten)	27,99€	Amazon
4.	Micro-USB Kabel			<i>vorhanden</i>	
5.	Dünnes Kabel		1 Meter	<i>vorhanden</i>	
6.	M3 14-20mm Schrauben			<i>vorhanden</i>	

## 6 Kosten

Wie der obigen Stückliste zu entnehmen ist, belaufen sich die initialen Kosten auf 75€, wobei zusätzlich noch ein Micro-USB Kabel, etwas Kabel zum Löten sowie gängiges Werkzeug wie Lötkolben, Lötzinn und ein M3 Schraubendreher benötigt werden.

Laufende Kosten fallen nahezu keine an, selbst im Fall eines Defektes sind beispielsweise noch genügend Ersatz-Tasten und Keycaps vorhanden, lediglich von den beiden Raspberry Pi Picos ist kein Ersatz vorhanden.

## 7 Zeitliche Planung

Kalenderwoche	Phase	Todo
15	Brainstorming	- Auf Idee festlegen
16	Bestellung	- Benötigte Hardware auflisten - Projektplanung a) Projektvorschlag Dokument b) Zeitplan erstellen c) Erstes Mockup für Layout - Hardware bestellen

17	Vorbereitung, PCB Design (rechts)	<ul style="list-style-type: none"> <li>- Git + GitHub Repo erstellen</li> <li>- KiCad einrichten und benötigte Komponenten kreieren               <ul style="list-style-type: none"> <li>a) Pi Pico</li> <li>b) Gateron Schalter</li> </ul> </li> <li>- Schematic erstellen</li> <li>- kleines Test-PCB entwerfen und Form davon 3D drucken (um Maße für Schalter zu prüfen)</li> </ul>
18		<ul style="list-style-type: none"> <li>- PCB Outline entwerfen</li> <li>- PCB Design fertigstellen</li> </ul>
19	PCB Bestückung, Test	<ul style="list-style-type: none"> <li>- PCB fertigen und bestücken</li> <li>- Funktionstest:               <ul style="list-style-type: none"> <li>a) jeden Schalter auslesen</li> <li>b) Tastatur emulieren mit Pi Pico</li> </ul> </li> </ul>
20	PCB Design (links)	<ul style="list-style-type: none"> <li>- Schematic (links)</li> <li>- PCB Design (links)</li> <li>- Fertigung des PCBs</li> </ul>
21		<ul style="list-style-type: none"> <li>- Bestückung links</li> <li>- Testen der linken Seite:               <ul style="list-style-type: none"> <li>a) jeden Schalter auslesen</li> </ul> </li> <li>- Verlöten des Kommunikationskabels zwischen Picos</li> </ul>
22	Kommunikation der beiden Hälften	<ul style="list-style-type: none"> <li>- Kommunikation zwischen Pico und Pico festlegen (PCB könnte alle):               <ul style="list-style-type: none"> <li>a) SPI</li> <li>b) I2C</li> <li>c) UART</li> <li>d) Custom</li> </ul> </li> <li>- sinnvolles Format für Kommunikation überlegen, um Buchstabe vom einen Pico zum anderen zu schicken (evtl. die anderen um Rat fragen)</li> </ul>
23		<ul style="list-style-type: none"> <li>- Code für jeweilige Seite entwickeln, der Tastendruck in Buchstabe umwandelt</li> </ul>

		(bzgl. genereller Architektur um Rat fragen)
24		<ul style="list-style-type: none"><li>- Buchstabe der linken Seite nach rechts schicken</li><li>- rechts kann mit empfangenen Buchstaben umgehen</li><li>- Emulation des Tastendrucks an PC</li></ul>
25	Gehäuse	<ul style="list-style-type: none"><li>- Design eines Gehäuses</li><li>- Bei genug Zeit: Handballenauflage</li></ul>
26		<ul style="list-style-type: none"><li>- Puffer (Shortcuts implementieren)</li><li>- Entwicklerdoku</li></ul>
27		<ul style="list-style-type: none"><li>Puffer (Präsentation erstellen)</li><li>- Benutzerdoku</li><li>- Veröffentlichung auf GitHub</li></ul>
28	Abschluss	<ul style="list-style-type: none"><li>- Vorstellung und Demonstration</li></ul>

## 8 Risiken

Das vermutlich größte Risiko am Projekt ist der Faktor Zeit, da ich mir nicht nur vorgenommen habe, ein neues Programm (KiCad) zu erlernen, sondern dies auch meine erste selbstgebaute Tastatur ist, was die Zeiteinschätzung im Vorfeld zusätzlich erschwert. Insbesondere der benötigte Software-Aufwand ist nur schwer einschätzbar, weshalb ich notfalls noch zwei Pufferwochen in den Zeitplan eingebaut habe (ließe sich im Extremfall weiter ausweiten, zum Beispiel durch Auslassen des Gehäuses und von Shortcuts). Natürlich hoffe ich trotzdem, das Projekt fristgerecht und vollständig fertigstellen zu können, damit es anschließend auch tatsächlich genutzt werden kann.