# 1. Einleitung

Unser Projekt im Rahmen des Moduls "Creative Technologies AG (CTAG)" im Wintersemester 2023 ist eine spannende interdisziplinäre Herausforderung. Es verbindet verschiedene Bereiche wie Technik, 3D-Konstruktion, 3D-Druck und Softwareentwicklung, um langfristigen Nutzen zu schaffen und die Schnittstelle zwischen diesen Disziplinen zu optimieren.

# 1.1 Hintergrund und Motivation

Unsere Motivation für dieses Projekt geht über die bloße Erfüllung einer Projektleistung hinaus. Wir möchten ein Projekt zu realisieren, das dauerhaften Wert bietet, sprich auch nach Abschluss des Moduls einen Mehrwert für uns bietet. Die Verbindung verschiedener Disziplinen ermöglicht es uns, innovative Lösungen zu schaffen und breites Fachwissen einzusetzen.

# 1.2 Projektziele

Unsere Ziele sind vielfältig. In einem ersten Schritt möchten wir die Arbeitsabläufe im bekannten Musikproduktionsprogramm FL Studio optimieren. Dies gewährleisten wir mithilfe von Makros und Tastenkombinationen, um die Produktivität der Benutzer zu steigern. Ebenfalls implementieren wir die Erweiterung unserer Lösung auf andere Softwareanwendungen, darunter Programme wie Illustrator, Visual Studio und weitere kreative Software mit komplexen Workflows.

# 1.3 Projektumfang

Der Umfang unseres Projekts erstreckt sich über die gesamte Prozesskette. Dies schließt die Konstruktion und den 3D-Druck des Gehäuses ein, gefolgt von der Beschaffung und Montage der technischen Komponenten. Zusätzlich umfasst unser Projekt die Programmierung und Konzeption der Makros sowie die Erstellung einer umfassenden Dokumentation. Diese Dokumentation gewährleistet, dass unser Projekt für andere zugänglich und nachvollziehbar ist.

# 1.4 Projektbeteiligte und Aufgaben

Unser Projektteam besteht aus mehreren Mitgliedern, von denen jeder spezielle Aufgaben und Verantwortlichkeiten übernimmt:

Marlon Noldin: Marlon ist verantwortlich für die Konzeption und Programmierung der Makros. Aimen Fehmi Ouzen: Aimen übernimmt die Rolle der Konstruktion, des Prototypings und des 3D-Drucks des Gehäuses.

# 2.Hardware:

## Printed Circuit Board:

Aufgrund unseres begrenzten Fachwissens im Bereich des PCB-Designs und der zeitlichen Einschränkungen aufgrund umfangreicher Projekte, haben wir uns dazu entschlossen, eine vorgefertigte PCB mit einem Layout von 12 Tasten und 2 Rotary Encodern zu verwenden. Diese PCB haben wir bei PCB-Way in Auftrag gegeben. (<https://www.pcbway.com/project/shareproject/DIY_Mechanical_Macro_Keypad_Ocreeb_24300065.html>)

Ein Bild, das Text, Screenshot, Elektronik, Schaltung enthält.

Automatisch generierte Beschreibung

Maße: 87mm x 81mm

## Große Bauteile:

### 12x Kailh Mechanical Switches:

Die Kailh Mechanical Key Switches, von denen wir insgesamt 12 verwendet haben, sind mechanische Tastenschalter von Kailh. Diese Tastenschalter zeichnen sich durch ihre Robustheit und Präzision aus. Wir haben eine 4x3 Layout für die Tasten benutzt.

(<https://www.digikey.ca/en/products/detail/adafruit-industries-llc/4952/14113453>)

Ein Bild, das Entwurf, Rechteck, Diagramm, Plan enthält.

Automatisch generierte Beschreibung

Abbildung ‑ Kahli Switch Fusion Abbildung

### 2x Rotary Encoder:

Wir haben uns für Standard-Drehknöpfe entschieden, die mit ihrer Knopf Funktion die 13. und 14. Tasten auf unserem Macro Pad repräsentieren. Diese Drehknöpfe zeichnen sich durch einen deutlichen Druckpunkt aus und bieten beim Drehen einen angenehmen Widerstand, was eine präzise Auswahl und Steuerung ermöglicht.

(<https://www.digikey.ca/en/products/detail/bourns-inc/PEC11R-4215F-S0024/4499665>)

Ein Bild, das Screenshot, Design enthält.

Automatisch generierte Beschreibung

Abbildung 0‑2 Rotary Encoder Fusion Abbildung

### 1x 0.96 Inch LCD Display mit Stemma Connector:

Das Display war eine spätere Ergänzung, da wir zunächst nicht wussten, dass der Microcontroller einen integrierten I2C Stemma Port hatte. Diese Entdeckung war äußerst praktisch, da sie die Notwendigkeit entfallen ließ, Beschriftungen für die variablen Tastenfunktionen zu erstellen. Das Display, das über den Stemma Connector angeschlossen wurde, bot mit seinen 128 x 32 Pixeln ausreichend Platz für unsere Zwecke und erleichterte die Bedienung des Macro Pads erheblich.

(<https://buyzero.de/products/monochrome-1-3-128x64-oled-graphic-display-stemma-qt-qwiic>)

Ein Bild, das Elektronisches Bauteil, Verbindungsstück enthält.

Automatisch generierte Beschreibung

Abbildung ‑ LCD Display von Adafruit Produktbild

## SMD-Bauteile:

### 1x Mikrocontroller: KB2040

Der KB2040 erfreut sich großer Beliebtheit bei der Erstellung von individuellen Tastaturen. Seine geringe Größe, die ausreichende Anzahl an adressierbaren Pins, die Verfügbarkeit eines USB-C-Anschlusses sowie des STEMMA I2C-Ports für Bildschirme machen ihn besonders attraktiv für die Entwicklung kleinerer Tastatur-Pads mit Bildausgabe. (https://www.adafruit.com/product/5302)

Ein Bild, das Elektronisches Bauteil, Elektrisches Bauelement, Elektronik, passives Bauelement enthält.

Automatisch generierte Beschreibung Ein Bild, das Text, Screenshot, Farbigkeit, Diagramm enthält.

Automatisch generierte Beschreibung

### 4x Neon Pixel:

Um unserem Marco Pad eine variable RGB-Underglow-Beleuchtung zu ermöglichen, haben wir vier Neo-Pixel verwendet. Die Neon Pixel erfordern eine Betriebsspannung von 5V. Da das KB2040-Board jedoch nur 3,3V ausgibt, war es notwendig, einen 5V-Level-Shifter in den Schaltkreis einzufügen, um die Neon Pixel korrekt zu betreiben. Sie sind auf der Unterseite der Platine in einem nach innen versetztem Viereck platziert um einen gleichmäßigen Underglow zu ermöglichen.

(<https://www.digikey.ca/en/products/detail/adafruit-industries-llc/1655/5154679>)

Ein Bild, das Screenshot, Kreis, Diagramm enthält.

Automatisch generierte Beschreibung

Abbildung ‑ Neon Pixel Fusion Abbildung

### 1x 5V Level Shifter:

Der 5V-Level-Shifter sorgt dafür, dass die vier Neon Pixel trotz der 3,3V-Spannungsversorgung, die er vom Marco Board über den PIN D.10 erhält, jeweils mit einer Spannung von 5V versorgt werden. Dies ermöglicht eine ordnungsgemäße Funktion der Neo-Pixel, da sie eine Betriebsspannung von 5V erfordern, selbst wenn die Steuerungseinheit des Marco Boards nur 3,3V ausgibt. Der Level-Shifter stellt sicher, dass die Spannung auf das erforderliche Niveau angehoben wird, um die Neon Pixel ordnungsgemäß zu betreiben.

(<https://www.digikey.ca/en/products/detail/texas-instruments/74AHCT1G125DBVTG4/1687957>)

Ein Bild, das Schwarzweiß, Design enthält.

Automatisch generierte Beschreibung mit mittlerer Zuverlässigkeit

Abbildung ‑ Level Shifter CAD Modell

### 12x Dioden:

Es wurden insgesamt 12 Dioden vor jeden Tastenkontakt gelötet, um dem Mikrocontroller die individuelle Ansteuerung jeder Taste zu ermöglichen und gleichzeitig das sogenannte "Ghosting" zu verhindern. Dieses Vorgehen gewährleistet eine präzise Erfassung der Tastenbetätigungen und verhindert, dass unbeabsichtigte Tastenanschläge oder Interferenzen zwischen den Tastenregistrierungen auftreten, was insbesondere bei Mehrfachbelegungen von Tasten wichtig ist.

(<https://www.digikey.ca/en/products/detail/smc-diode-solutions/1N4148W/6022450>)

Ein Bild, das Screenshot, Design enthält.

Automatisch generierte Beschreibung

Abbildung ‑ Dioden Fusion Abbildung

### 4x Kondensatoren:

Vor jedes der vier Neo-Pixel wurden Kondensatoren geschaltet, um eine stabile Spannungsversorgung sicherzustellen und mögliche Spannungsspitzen zu puffern. Diese Kondensatoren dienen dazu, Schwankungen in der Stromversorgung zu minimieren und die Zuverlässigkeit der Neo-Pixel sicherzustellen. Da diese um die verschiedenen Farben darzustellen auf sehr feine Spannungsschwankungen reagieren.

(<https://www.digikey.ca/en/products/detail/stackpole-electronics-inc/RMCF0805JG10K0/1757769>)

Ein Bild, das Screenshot, Design enthält.

Automatisch generierte Beschreibung

Abbildung ‑ Kondensatoren

### 

### 1x 10 kOhms Resistor:

Der eine Widerstand ist vor dem Level-Shifter geschaltet, um die korrekte Spannungsanpassung und den Schutz des Level-Shifters sicherzustellen. Dieser Widerstand spielt eine wichtige Rolle bei der Aufrechterhaltung der richtigen Signalpegel und verhindert mögliche Schäden am Level-Shifter durch einen übermäßigen Stromfluss. Dadurch wird eine zuverlässige Signalübertragung zwischen dem Mikrocontroller und dem Level-Shifter gewährleistet.

(<https://www.digikey.ca/en/products/detail/stackpole-electronics-inc/RMCF0805JG10K0/1757769>)

Ein Bild, das Screenshot, Text, Design enthält.

Automatisch generierte Beschreibung

Abbildung ‑ Resistor Fusion Abbildung

## Bestellvorgang und Lieferung:

Für die PCBs haben wir uns für einen chinesischen PCB-Hersteller entschieden, da sie das beste Preis-Leistungs-Verhältnis in Bezug auf die Lieferzeit geboten haben. Dies ermöglichte uns, die Kosten niedrig zu halten und dennoch eine akzeptable Lieferzeit zu gewährleisten.

Bei den Bauteilen haben wir einen Lieferanten gesucht, der alle benötigten Komponenten anbietet, um Versandkosten zu sparen und die Bestellabwicklung zu vereinfachen. Wir haben außerdem darauf geachtet, von jedem Bauteil ausreichend Mengen zu bestellen, um im Notfall Ersatzteile zum Löten zur Verfügung zu haben. Dieser Ansatz ergab Sinn, da die Kosten für die Bauteile selbst vergleichsweise niedrig waren und die Versandkosten einen Großteil der Gesamtkosten ausmachten.

## SMD-Löten:

Der Lötvorgang und das SMD-Verfahren stellten während des Projekts eine spannende Herausforderung dar, insbesondere für mich, da ich zuvor keinerlei Erfahrung im Löten hatte. Besonders die Verwendung eines Lötkolbens mit großen Spitzen gestaltete sich anfangs schwierig, insbesondere wenn es um die präzise Verbindung der kleinen SMD-Komponenten auf der PCB ging. Die winzigen Bauteile erforderten höchste Präzision und Geduld, und es war eine steile Lernkurve zu bewältigen.

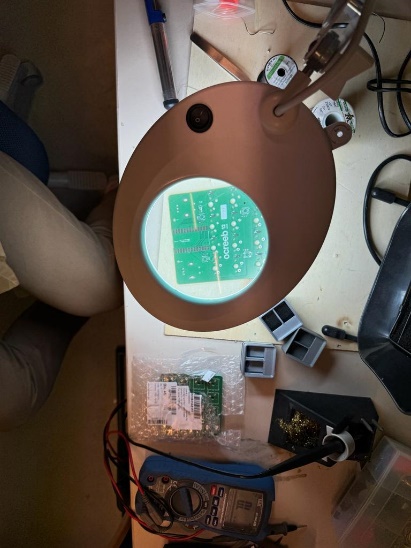
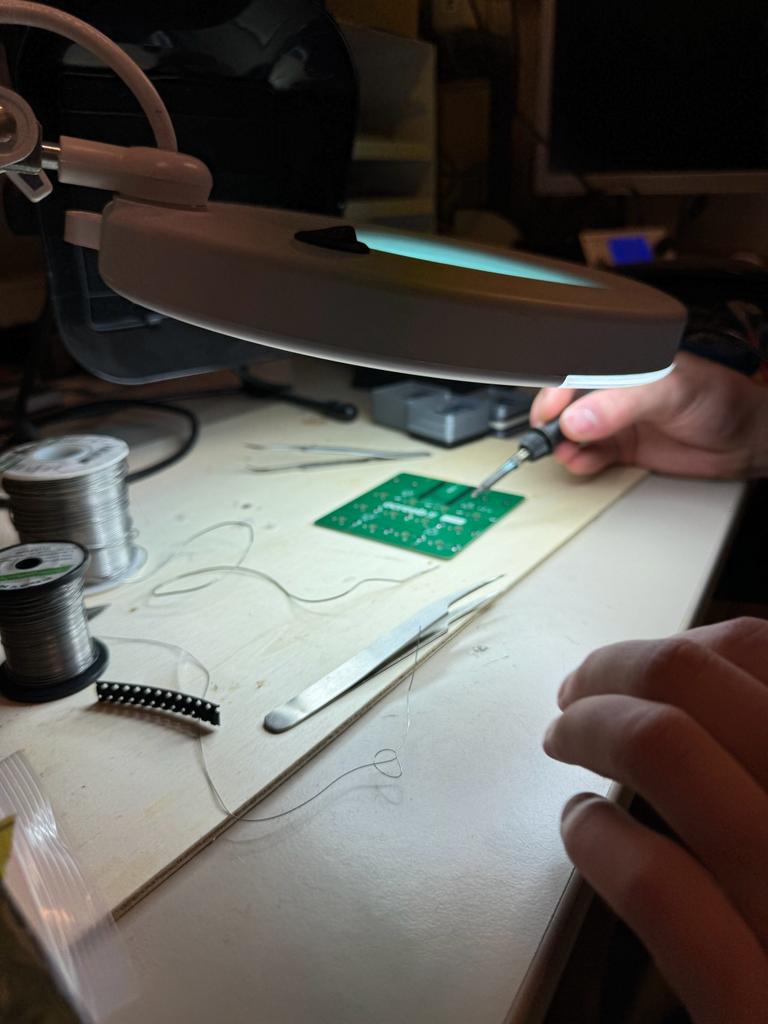
Ein wichtiger Moment war, als ich etwa die Hälfte der Lötarbeiten im Maker Space an der FH Kiel durchgeführt habe. Dort erhielt ich Unterstützung von den Hiwis (,die mir wertvolle Tipps und Anleitungen beim Löten der SMD-Teile gaben. Trotz meiner anfänglichen Unerfahrenheit in diesem Bereich konnte ich durch ihre Hilfe meine Fähigkeiten entwickeln und zur erfolgreichen Umsetzung des Projekts beitragen.

Dennoch wurde mir im Laufe des Projekts klar, dass ich häufiger Zugang zu einem Lötkolben benötigte, da ich in Rendsburg wohne und der Weg nach Kiel nicht immer praktikabel war. Daher entschied ich mich, einen eigenen Lötkolben für den Heimgebrauch anzuschaffen. Dies ermöglichte mir, die Lötarbeiten flexibler zu gestalten und sie bequem in meiner eigenen Umgebung durchzuführen.

Es ist erwähnenswert, dass es während des Projekts eine Situation gab, in der wir eine PCB aufgrund einer fehlerhaften Verlötung verwerfen mussten. Dies unterstreicht die Herausforderungen, die mit dem Löten von SMD-Komponenten verbunden sind, und betont die Bedeutung von Präzision und Sorgfalt während des Lötprozesses. Trotz meiner mangelnden Erfahrung zu Beginn des Projekts und dieses Zwischenfalls verliefen die Lötarbeiten insgesamt erfolgreich, und wir konnten unsere Ziele für das Projekt erreichen.

Zum Ende hin hat sich nur herausgestellt das die FH mit dem Lötofen eine viel einfachere Möglichkeit hatte SMD Bauteile zu löten.

*Hier Bilder zu unserem Löt-Werdegang:*

# Software:

## Entwicklungsumgebung:

Bei der Wahl der Entwicklungsumgebung hatten wir die Option, zwischen der weit verbreiteten Arduino-Entwicklungsumgebung und CircuitPython zu wählen. Arduino ist bekannt für seine Standardisierung im Bereich der Mikrocontroller-Programmierung, während CircuitPython die Verwendung von normalem Python auf dem Controller ermöglicht. Da wir bereits Erfahrung mit Python aus dem Modul Cyberphysical Systems hatten und keine Vorkenntnisse in C# besaßen, fiel unsere Wahl letztendlich auf CircuitPython.

Ein entscheidender Faktor bei dieser Wahl war auch die Verfügbarkeit der KMK (Key Macro Keyboard) Library für Python. Diese Library war essenziell für die Implementierung unserer Makrofunktionen und ermöglichte uns, die gewünschten Funktionalitäten effizient umzusetzen. Durch die Verwendung von CircuitPython konnten wir sicherstellen, dass unser Macro Pad optimal den Anforderungen entspricht und gleichzeitig unsere bereits vorhandenen Python-Kenntnisse nutzen.

## KMK Library:

Die KMK Library ist eine umfangreiche Bibliothek, die eine Vielzahl von Möglichkeiten bietet, um ein individuelles Keyboard zu erstellen. Für unsere Zwecke erwies sich diese Library als äußerst nützlich. Mit ihr konnten wir nicht nur die Neon-Pixel und das Display steuern, sondern auch die Layer-Funktion nutzen. Diese Funktion ermöglichte es uns, jeder Taste eine bestimmte Funktion zuzuweisen, die sich beim Wechseln der Layer änderte. Darüber hinaus erlaubte uns die Library das Auslösen verschiedener Funktionen durch Doppel- oder Dreifachtippen auf Tasten. Ein weiterer Vorteil war die Unterstützung von Rotary Encodern, die in unserem Projekt ebenfalls eine wichtige Rolle spielten. Insgesamt bot uns die KMK Library die notwendigen Werkzeuge und Funktionen, um sehr einfach und übersichtlich ein leistungsstarkes und individuell anpassbares Macro Pad zu entwickeln. Zusätzlich ermöglichte sie die Simulation von Maustasten und des Mausrads, was die Vielseitigkeit unseres Projekts weiter erhöhte.

## Automatischen auswählen der Layer:

Die Fähigkeit des Microcontrollers, über USB-C mit dem PC zu kommunizieren, ermöglichte die Implementierung eines Python-Skripts. Dieses Skript verfolgt die geöffneten Programme auf dem PC und wechselt automatisch den Layer des Macro Pads, wenn ein Programm einem bestimmten Funktionssatz oder Layer zugeordnet ist. Dies wurde realisiert, indem der Fenstertitel des aktuell fokussierten Programms analysiert wurde. Wenn der Titel zu einem der vordefinierten Layer passte, wurde der entsprechende Layer-Index an den Macro Controller gesendet. Dieser setzte dann die Tastenbetätigung entsprechend um und wählte den gewünschten Layer aus.

Da die Layer jeweils zu verschiedenen Anwendung zugehörig sind. Ermöglicht dieses Skript die Nutzung des Macro Pad noch viel Intuitiver zu machen da man selber nicht mehr aufpassen muss auf welchen Layer man ist.

Ein Bild, das Text, Screenshot, Software enthält.

Automatisch generierte Beschreibung

Abbildung ‑ Script auf dem Rechner für die festlegung des Layers

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung ‑ Die Verschieden Farben für die Layer setzen und Die Layer switchen

## Keyboard Initialisieren

Die unten gezeigte Python-Code-Initiierung ist ein Beispiel für das grundlegende Setup für das Programmieren eines Macro-Pads unter Verwendung des KMK-Frameworks (Keyboard Maintainer's Kit). Lassen Sie uns die wichtigen Teile des Codes erläutern:

1. Import- Anweisungen:

Hier werden die erforderlichen Python-Module und Bibliotheken importiert, die für die Programmierung des Macro-Pads benötigt werden. Dies umfasst das board-Modul, das customKeys-Modul und verschiedene Module aus dem KMK-Framework, wie das OLED-Display, Tasten, Layer-Management, Encoder, Medientasten, TapDance und Maustasten.

1. Keyboard-Setup:

Dieser Abschnitt konfiguriert das Keyboard und seine verschiedenen Module. Es erstellt eine Instanz des KMK-Keyboard-Objekts und fügt verschiedene Module hinzu, darunter Layers (Ebenen), Encoder-Handler, TapDance (für spezielle Tastendrucksequenzen), MouseKeys (für Maussteuerung) und MediaKeys (für Mediensteuerungstasten).

1. Tasten- und Encoder-Pins:

Hier werden die Pins für die Tasten und Encoder festgelegt, die mit dem Macro-Pad verbunden sind. Dies ermöglicht die physische Verbindung und Interaktion mit den Tasten und Encodern.

1. Diode Orientation:

Diese Zeile legt die Orientierung der Dioden auf der Platine fest. In diesem Fall wird die Orientierung auf "COL2ROW" gesetzt, was bedeutet, dass die Dioden in einer Spalte und zu den Zeilen hin ausgerichtet sind. Dies ist wichtig für das ordnungsgemäße Funktionieren der Tastenmatrix.

Ein Bild, das Text, Screenshot, Software enthält.

Automatisch generierte Beschreibung

Abbildung ‑ Imports und Setup fürs Macropad

## Marcos erstellen:

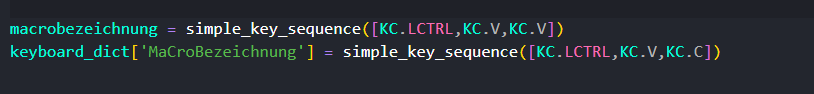


Abbildung ‑ Macros erstellung

Dieses Beispiel zeigt, wie man ein Macro in einem Code-Beispiel erstellt:

**simple\_key\_sequence:** Diese Funktion ermöglicht die Erstellung einer Sequenz von Tastenanschlägen, die nacheinander ausgeführt werden. In diesem Fall umfasst die Sequenz:

**KC.LCTRL:** Die linke Strg-Taste wird gedrückt.

**KC.V:** Dann wird die Taste "V" gedrückt, was normalerweise die Einfügeoperation auslöst.

**KC.MACRO\_SLEEP\_MS(1000):** Hier wird eine Wartezeit von 1000 Millisekunden (1 Sekunde) eingefügt. Dies kann nützlich sein, um sicherzustellen, dass die folgenden Aktionen nicht zu schnell ausgeführt werden.

**send\_string("Hello World"):** Schließlich wird der Text "Hello World" gesendet, der in das aktive Programm oder Fenster eingefügt wird.

Die send\_string-Funktion ist nützlich, um Text automatisch einzugeben, was beispielsweise hilfreich sein kann, wenn du häufig denselben Text in Dokumenten oder Anwendungen eingeben musst. Leider kann die sendString() Methode nur im US Layout tippen deshalb muss das im Endeffekt umgekehrt werden. Der KC.MACRO\_SLEEP\_MS-Befehl ermöglicht es dir, Verzögerungen zwischen den Tastenanschlägen zu steuern, was wichtig sein kann, um sicherzustellen, dass die Eingaben korrekt erkannt werden, insbesondere in Anwendungen, die auf Tastenfolgen empfindlich reagieren. Makros können in verschiedenen Szenarien nützlich sein, von der Textautomatisierung bis hin zur schnellen Ausführung von Befehlen oder Aktionen in Softwareanwendungen. Sie sparen Zeit und reduzieren die Notwendigkeit, wiederholt denselben Text oder dieselben Tastenkombinationen manuell einzugeben.

## Layer erstellen und mit Macros befüllen:

Die Layer werden in einer zweidimensionalen Map abgebildet, wobei die verschiedenen Tasten für die Layer in der zweiten Dimension durch Kommas getrennt sind. Dies ermöglicht es, verschiedene Layer und deren Tastenbelegungen klar zu definieren und zu organisieren.

Die Verwendung der KC.TO(0)Taste ist eine Möglichkeit, zwischen den verschiedenen Layern zu wechseln. Wenn die KC.TO(1)Taste gedrückt wird, erfolgt ein Layer-Wechsel gemäß der in der Layer-Map definierten Konfiguration. Dies ermöglicht es dem Benutzer, je nach Bedarf auf unterschiedliche Tastenbelegungen und Funktionen zuzugreifen, indem er einfach zwischen den Layern hin- und herschaltet.

Ein Bild, das Text, Screenshot, Schrift, Software enthält.

Automatisch generierte Beschreibung

Abbildung ‑ Key Belegung der innerhalb der Layer

## Display ansteuern:

Um Abkürzungen für die jeweiligen Tasten auf unserem LCD-Display darzustellen, haben wir eine kreative Lösung entwickelt. Wir verwendeten ein Dictionary, in dem die Tastenfunktionen als Schlüssel (Key) und ihre Beschreibungen als Werte (Value) gespeichert wurden. Dann haben wir die Keymap-Liste durchlaufen, um die Beschreibung für jede Taste zu finden.

Der Prozess war wie folgt:

1. Wir haben ein Dictionary erstellt, in dem die Tastenfunktionen als Schlüssel und die dazugehörigen Beschreibungen als Werte gespeichert wurden.
2. Anschließend haben wir die Keymap-Liste durchiteriert, um die jeweilige Funktion jeder Taste zu ermitteln.
3. Wir haben die gefundene Funktion als Schlüssel im Dictionary verwendet, um die dazugehörige Beschreibung (Value) zu finden.
4. Die Beschreibung wurde dann in einen String umgewandelt, aus dem wir die Großbuchstaben herausgefiltert haben. Diese Großbuchstaben wurden schließlich als Abkürzung auf dem LCD-Display angezeigt.
5. Diese Methode ermöglichte es uns, die Tastenbeschriftungen dynamisch basierend auf den Funktionen und ihren Beschreibungen zu generieren, ohne jede Beschriftung manuell eingeben zu müssen. Dies trug zur Benutzerfreundlichkeit unseres Macro Pads bei und erleichterte die Identifizierung der Tastenfunktionen erheblich.

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung ‑ Umwandlung zur Keymap

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung ‑ Speicherung der Keywerte als Dictionary im Keymap Layer

# 4. Modellierung

Um unsere Komponenten verstauen zu können, habe ich mich an das Design des Macro Pads ran gesetzt.

Hierbei musste ich vieles beachten. Zum einen, dass die Tasten und Switches den gleichmäßigen und richtigen Abstand haben, um nicht miteinander anzuecken, die Kabelführung von der Platine zum Display (dazu später mehr) funktioniert, ohne Verluste aufgrund der Kabellänge. Und zum anderen, dass die sowohl Bottom Case und Top-Case einen Verschluss haben ohne Spiel. Zu Beginn habe ich in Blender den ersten Prototypen modelliert, um diesen dann in Fusion 360 zu erstellen.

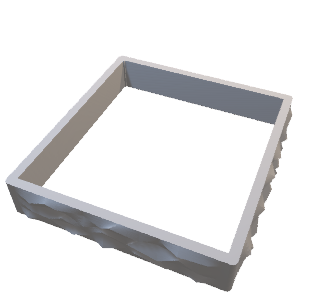
## 1. Iteration:

Ein Bild, das Quadrat, Screenshot, Tablet, Design enthält.

Automatisch generierte BeschreibungUnser Fokus bei dem ersten Modell lag bei den richtigen Bemaßungen und dem richtigen Verschluss. Daraufhin haben wir das erste Modell mithilfe des 3D-Druckers gedruckt. Nachdem wir festgestellt haben das alles so weit passte haben wir uns darüber Gedanken gemacht, wie wir von dem simplen Design wegkommen können.

## 2. Iteration:

Bei der 2ten Iteration lag unser Fokus beim Design des Fraktalmusters, hier haben wir unter Fusion mit dem Knicken der Formen gearbeitet. Schnell wurde uns klar, dass es ersten sehr cool aussieht aber wahrscheinlich nicht so „schön“ aussehen könnte, wenn es aus dem 3D Drucker kommt.

Zudem haben wir die Aussparung der Tasten vertieft, um die Keys nicht allzu weit oben stehen.  
Gedruckt haben wir es aber erst in der 3ten Iteration.

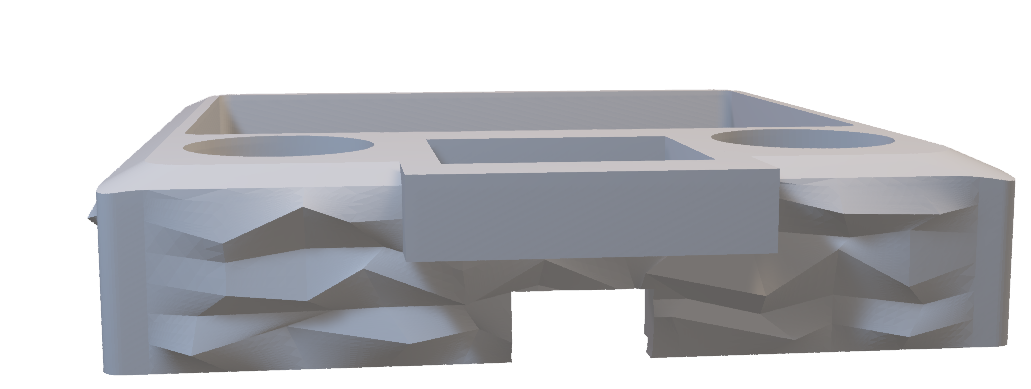
## 3. Iteration:

In der 3ten Iteration lag unser Fokus beim Display. In der allerersten Planung war zwar die Rede davon, aber wir wussten vorher nicht, wie wir das Display ansteuern würden, und wo sich die Stromversorgung dafür befindet. Deswegen haben wir uns erst zum späteren Zeitpunkt mit der Implementierung des Displays im Modell beschäftigt. Hier war klar, dass wir Platz für das Kabel machen mussten, weswegen das „Modul“, in dem sich das Display befindet schräg / angewinkelt war, da wir nicht wussten, ob die langen Stecker mit dem Gehäuse kollidieren. Da wir manchmal Schwierigkeiten hatten, unsere Kabel mit dem USB C Port des Boards zu verbinden haben wir uns auch dazu entschieden das Loch für diesen etwas größer zu gestalten. Deswegen haben wir mit eher zu viel Luft gearbeitet als mit viel zu wenig. Dies haben wir dann mit den Änderungen aus der 2ten Iteration auch gedruckt.

## 4. Iteration:

Nachdem wir das richtige Display hatten (mit dem richtigen dazugehörigen Stecker), haben wir uns dazu entschieden die schräge Anordnung für das Display zu verwerfen und das Display einfach horizontal zu montieren ohne Herausrangenden „Modulen“. Der Vorteil hierbei war, dass die Stützen minimal sind, und somit der Prozess des entfernen der Stützen uns teilweise erspart bleibt.

Nach dem Abschließen der 4ten Iteration haben wir unser Finales-Design gehabt und waren auch sehr zufrieden damit.

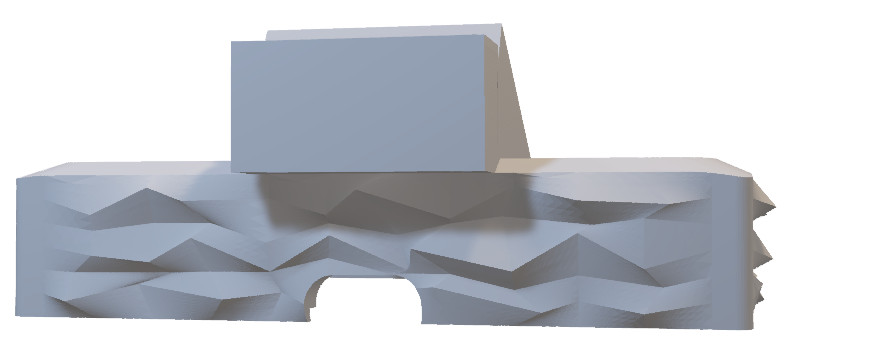


## 1. Prototyp:

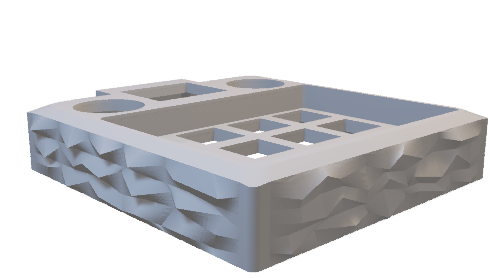
Unser erster Prototyp zeigt den Verschluss Mechanismus, um das Gehäuse schließen zu können. Dieser durfte aber auch nicht zu wenig spiel haben, damit die Komponenten auch nicht rausfallen.



## 2. Prototyp:

Unser zweiter Prototyp umfasst die zweite und dritte Iteration.  
Das heißt, dass wir in diesen Prototypen die schräge / angewinkelte Aussparung für das Display haben. Zudem haben wir auch die etwas tiefere Aussparung für die Tasten und Switches.

## 3. Finaler Prototyp:

In dem letzten und Finalen Prototypen haben wir das Fraktal Muster, mit der Vertiefung für die Tastatur und die etwas größere Aussparung für den USB C Ports um das Reibungslose verbinden zu garantieren. Wir haben auch hier das horizontale Design umgesetzt für das Display.

# Fazit:

Ja, im Endeffekt sind wir sehr froh, uns für diese Semesteraufgabe in CTAG eine solche Herausforderung gesetzt zu haben. Sie hat uns an manchen Stellen stark gefordert, aber auch tiefe Einblicke in verschiedenste Branchen und Bereiche ermöglicht. Über den Verlauf des Projekts gab es oft die Befürchtung, nicht rechtzeitig zur Abgabe fertig zu werden. Deshalb sind wir glücklich, es kurz vor knapp geschafft zu haben und zur Vorstellung einen funktionstüchtigen Prototypen präsentieren zu können. Das Projekt hat nicht nur Spaß gemacht, sondern wurde auch mit viel Ehrgeiz bearbeitet. Angesichts der potenziellen Nutzbarkeit am eigenen PC könnte es sehr nützlich sein und ist kein Projekt, das man nach dem Semester einfach vergisst.

# Ausblick in die Zukunft:

Dank der vielfältigen Möglichkeiten, die die KMK Library bietet, wird in Zukunft wahrscheinlich daran gearbeitet, die verschiedenen Macros verstärkt in unsere Arbeitsroutine zu integrieren. Darüber hinaus wird noch untersucht, wie das Skript zur Steuerung der Layer beim Start des PCs automatisch gestartet und im Hintergrund ausgeführt werden kann. Auch die Ansteuerung der RGB-Beleuchtung ist noch nicht optimal gelöst, daher werden wir prüfen, ob zusätzliche Effekte und Animationen programmiert werden können.