

# Schachcomputer mit Sensorboard und Magnetverschiebung



## Konzept

### Allgemein

Keine Bedienelemente, nur ein Spielbrett mit Ein-/Ausschalter und USB-Buchse.

Testroutinen werden durch das Setzen von Figuren auf Reihe D vor dem Einschalten aktiviert.

Über den USB Eingang wird das System versorgt und ein integrierter Puffer-Akku geladen. Zudem dient er als Programmierschnittstelle. Betrieb ohne Netzteil ist für ca. 15 Minuten möglich. Dann geht stoppt das Spiel und zeigt den Fehler „Akku leer“ mit abwechselndem rotem Blinken an. Nach Anschluss einer Spannungsversorgung, wird das Spiel nach ca. 5 Minuten fortgeführt.

### Bedienung

Normales Spiel Mensch gegen Computer, wenn beim Start alle Figuren am Platz stehen.

Wird in der Startaufstellung der linke weiße Turm abgehoben und wieder gesetzt, beginnt der automatische Demomodus Computer gegen Computer. Der rechte weiße Turm aktiviert in gleicher Weise den Seitentausch und der Spieler spielt die schwarzen Figuren.

Dreimaliges heben und absenken des weißen Königs startet die automatische Neuaufstellung.

Vom Computer geschlagene Figuren werden zum oberen Rand gefahren und dort abgestellt. Der Spieler muss die von ihm geschlagenen Figuren selbst vom Brett nehmen.

Die Züge müssen klar gesetzt und genommen oder verschoben werden.

Wichtig ist komplexe Züge, wie z.B. das Schlagen oder die Rochade Schritt für Schritt auszuführen. So ist eine klare Erkennung des durchgeführten Zuges möglich.

Bei verbotenen Zügen erfolgt ein Warnton.

## Statusanzeige

| Akku-Led | Status-Led | Bedeutung |
|----------|------------|-----------|
|----------|------------|-----------|

### Initialisierungs-codes und Störungen

|            |            |   |
|------------|------------|---|
| Hellblau   | Hellblau   | Spannung liegt an, System noch nicht initialisiert, Fehler in boot.py |
| Gelb       | Gelb       | Gehäuse geöffnet, System angehalten                                   |
| aus        | Gelb       | Diagnose aktiv  |
| aus        | Rot        | Fehler beim Schnelltest, Versuche Neuinitialisierung und Neustart     |
| Rot        | Rot        | Panik-Modus, USB Fehlerausgabe, System angehalten                     |
| Rot/Aus 1s | Aus/Rot 1s | Akku leer, wechselseitiges blinken, System pausiert                   |

### Generelle Benutzerführung

|      |      |  |
|------|------|--|
| Gelb | Grün | Spielbrett noch nicht komplett aufgebaut (Beep wenn ok)                      |
| *    | Grün | System wartet auf Aktion des Spielers  |
| *    | Gelb | Erfassung Zug des Spielers begonnen<br>im Demomodus: Zug Weiß wird berechnet |
| *    | Rot  | Zug des Systems wird berechnet<br>im Demomodus: Zug Schwarz wird berechnet   |

### Aktueller Ladezustand des Akkus wird während des Spiels angezeigt

|      |   |  |
|------|---|--|
| Grün | * | Akku ausreichend geladen                     |
| Gelb | * | Akku <66%, begrenzte Restlaufzeit            |
| Rot  | * | Akku <33%, laden, bitte Netzteil anschließen |

### Abschluss des Spiels

|             |             |   |
|-------------|-------------|---|
| Grün/Aus 1s | Grün/Aus 1s | Spiel beendet, nach 1min automatischer Neustart des Systems |
|-------------|-------------|---|

## Testroutinen

Die Testroutinen werden durch spezielle Figurenstellungen beim Start aktiviert.

### Systemtests

D1: Akkustand anzeigen, Figur fährt von A1 (leer) bis A8 (voll), je nach gemessener Akkuspannung

D4: Parkposition Magnet, wenn nicht besetzt erfolgt beim Systemstart automatische Feinkalibrierung

D7: Testroutine Mechanik und Sensorik, Figur wird auf jedem Feld kurz vermessen

Bei Fehler wird Figur stehengelassen, wenn alles in Ordnung auf der Halde abgelegt.

Achtung: Hohe thermische Belastung des Magneten, nicht öfters hintereinander ausführen!

D8: Initialisierung der Bewegungsmechanik auslösen.

## Software

### Engine

MicroPython hat deutliche Stärken in der Listenverarbeitung und dem Entwicklungskomfort. Durch den Cross-Compiler werden Speicherprobleme beim Laden der Module umgangen. Zur Laufzeit ist die Speicherauslastung dann absolut unkritisch.

In C++ ist die Implementierung wesentlich effizienter, der Entwicklungsaufwand aber deutlich höher. Der kompilierte C++ Code läuft wesentlich schneller als Python Skripte und braucht weniger Speicher.

Im ersten Ansatz wurde MicroPython der Vorzug gegeben, da zuerst grundlegende Konzepte erarbeitet werden sollen und die Entwicklungsperformance im Vordergrund steht.

### Bewegungssteuerung

Um zwischen zwei Figuren passieren zu können, ohne diese zu verschieben, wird viel Platz benötigt. Die Figur darf einen maximalen Durchmesser von 50% der Feldbreite haben.

Um dies auf 75% zu steigern, muss man die benachbarten Figuren an den Rand schieben. Die Figuren werden dabei jeweils um 25% der Feldbreite zur Seite geschoben.

Durch die Beschränkung der X-Achse kann aus A nicht nach links und aus H nicht nach rechts ausgewichen werden. Hier ist der Platz durch Verschieben von B+C bzw. F+G einseitig zu schaffen. Ebenso kann in Reihe 1 nicht nach unten verschoben werden.

Spielunterbrechungen sind problematisch, da Figuren technisch nicht identifiziert werden können. Der Akku erlaubt jedoch kurzzeitiges Abklemmen der Spannungsversorgung ohne Probleme.

Die Navigation der Figuren auf dem Schachbrett, also das Ausführen von Zügen, erfolgt mittels eines Flooding-Algorithmus. Dieser bezieht die belegten Felder, sowie die Ecken und Kanten mit ein um das Umfahren von Figuren zu ermöglichen.

Die HAL-Sensoren streuen stark, hier ist ein individueller Abgleich der Messung erforderlich. Dafür ist die direkte Steuerung der Schrittmotoren mit MicroPython einfach und problemlos möglich.

### Benutzerinterface

Die Anforderung an eine einfache Bedienung und der Verzicht auf jegliche Steuerelemente erfordert eine präzise Analyse der jeweiligen Statusübergänge. Hierzu wurde eine State-Machine implementiert. Jede Aktion führt zu einem entsprechenden Statuswechsel.

Die Rücknahme von Zügen ist zumeist durch einfaches Zurückstellen möglich.

### Schachprogramm

Das eigenentwickelte Schachprogramm basiert auf dem Min/Max-Algorithmus.

Dieser ist als rekursiver Negamax mit nachgeschalteter Quiescence-Suche implementiert. Zur Bewertung werden die Figuren und deren Stellung individuell gewertet.

Zur Reduktion des Suchbaums ist ein Alpha/Beta cut-off mit Vorsortierung der Züge realisiert.

Die Entwicklung erfolgte parallel zum Bau der Mechanik, viele Details wurden ausprobiert und verworfen.

Die Ressourcen des Microcontrollers mit MicroPython sind ausreichend um später noch weitere Verbesserungen zu erreichen.

## Hardware

### Aufbau

Zentrales Spielfeld mit 200 mm, Rand 50 mm, Feldgröße 25 mm, Figurendurchmesser bis 20 mm.  
Über dem Spielfeld Halde für ca. 10 vom Computer geschlagene Figuren, knapp, aber ok.

Status LED an der Front.

Ein/Aus-Schalter, USB-Anschluss für Programmierung und als Netzteilanschluss an der Rückseite.

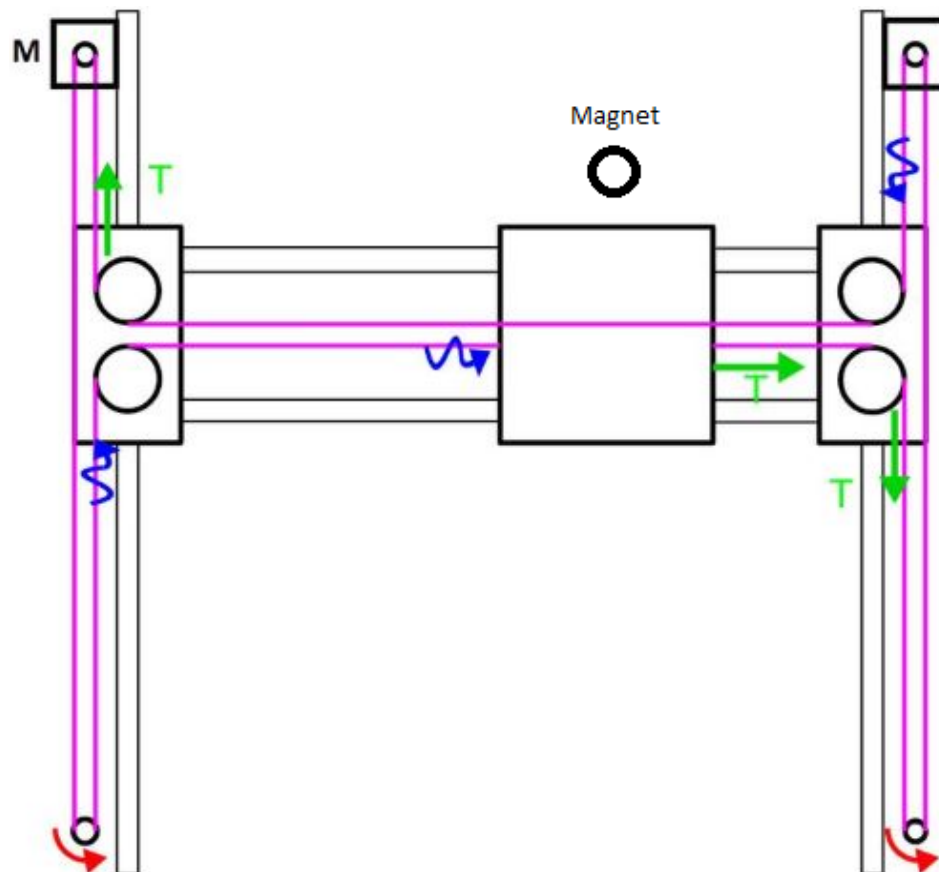
Über USB wird ein LiPo Akku geladen, der dann die Stromspitzen bei Aktivierung des Magneten und der Schrittmotoren abfängt. Zudem dient der USB-Anschluss als Programmierschnittstelle.

### XY Tisch

Geringe Kräfte und geringe Anforderung an Genauigkeit +/-1mm sind genug.

Verwendung einer vorhandenen 220mm und zweier 270mm THK MGN12 Linearführungen.

Der Aufbau basiert auf dem einfachen H-Bot Prinzip:



Im Vergleich zum komplexeren CoreXY Design neigt der Schlitten zum Verdrehen unter Last.

Dies ist aber kein Problem, da keine nennenswerten Kräfte übertragen werden.

Zudem benötigt CoreXY im Vergleich zu H-Bot mehr Bauhöhe, da sich die Zahnriemen kreuzen.

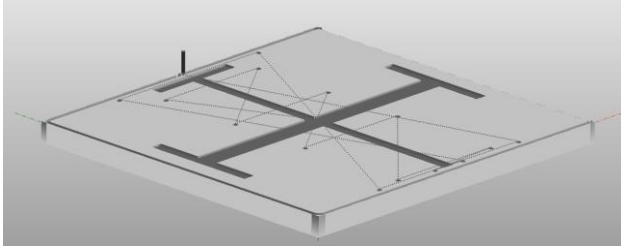
Die Motoren sind oben montiert, da dort durch den erforderlichen Versatz des Elektromagneten näher an den Rand gefahren werden kann. Somit kann die Halde für geschlagene Figuren realisiert werden. Der Versatz ist erforderlich, da sonst der Magnet dem Zahnriemen im Weg ist.

Der Fahrweg X-Achse beträgt 179mm und in der Y-Achse 201mm bei 291x291mm Grundfläche.

## Aufbau

### Grundplatte

Für die Maßhaltigkeit am besten verzugsfreie 16mm Faserplatte, 35x35cm als Basis.  
Der Korpus über Auflagern auf der Grundplatte und ist vorne und hinten 2fach verschraubt.  
In Längsrichtung bieten die Linearführungen die notwendige Steifigkeit.



Der Platz für die Kontrollplatine und die Leistungsplatine ist begrenzt.  
Der Aufbau auf den Platinen ist auf 25mm Höhe begrenzt.  
Der Rauminhalt wird praktisch vollständig benötigt.

### Brettmaße

Feldgröße 25x25mm  
Figuren 20mm

### Höhenmaße

#### Führung

Y-Führungen 13mm  
Y-Halter 3mm  
X-Führung 13mm  
Wagen 4mm (unter Magnet)  
Magnet 20mm

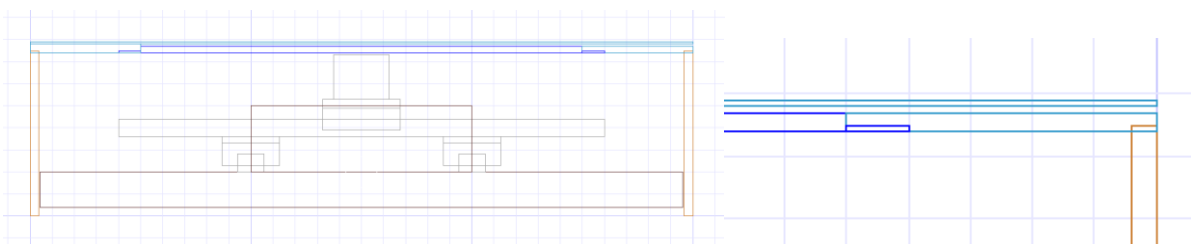
Somit Magnetoberkante 53mm über Grundplatte.

### Schrittmotoren/Zahnriemen

Die flachen Nema17 Schrittmotoren erfordern eine Bauhöhe von mind. 45mm.  
Der 6mm Zahnriemen läuft auf 41-48mm Höhe (1mm Spiel).  
Die Y-Umlenkrollen sind 1mm höher montiert damit die Umlenkrollen der X-Achse nicht kollidieren, sondern eintauchen, falls bis Anschlag gefahren wird.

### Festlegung der Bauhöhe:

Höhe der Bewegungsmechanik ist auf 55mm Abstand Grundplatte zum Deckel abgestimmt.  
Magnetoberkante bei 53mm, 1mm Luft zur Unterkante Deckplatte.  
1mm Aufbau für das abnehmbare Sensorboard. Diese benötigt 3mm, wird aber versenkt eingebaut.  
Im Ausschnitt des Sensorboards ist das Furnier mit einer 1mm GFK-Platte verstärkt.  
Hierdurch werden hauptsächlich Verwindungen des Furniers verhindern.  
Das Spielfeld ist dann auch nicht mehr so nachgiebig und fühlt sich wertiger an.

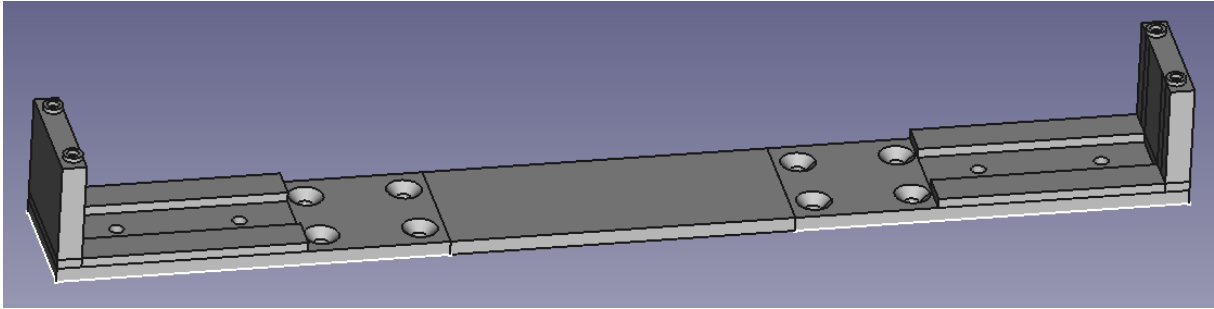


## 3D-Druckteile

### X-Bruecke

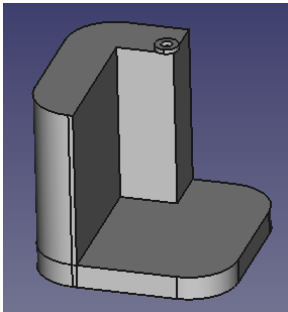
Verbindet die X-Achse mit den zwei Schlitten der Y-Achsen.

Deutlich Verwindungssteifer als die ursprünglich geplanten zweiteiligen X-Halter.



### Y-Umlenkrolle

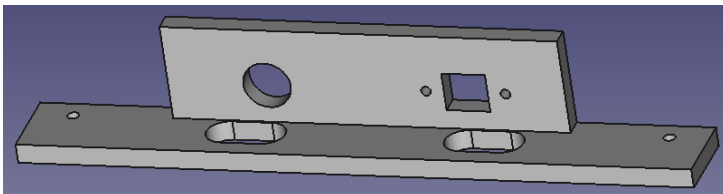
Gegenstück zu dem Schrittmotoren, lenkt den Zahnriemen links und rechts vorne um.



### Anschlussblende

Erlaubt die Befestigung der Anschlüsse an der Grundplatte statt am Korpus.

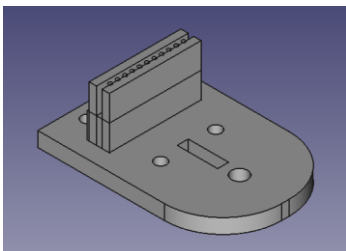
Die Blende kann z.B. für einen zusätzlichen Anschluss ohne größere Umbauten erweitert werden.



### Magnethalter

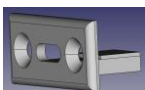
Koppelt den Magnetschlitten an den Zahnriemen und ist die Auflage des Magneten.

Der Mitnehmer klemmt den GT2 Zahnriemen ein, hierfür ist keine Verschraubung erforderlich.



### USB-Buchsenhalter

Hiermit wird die USB-Buchse an der Anschlussblende verschraubt.



## Entwurf der Elektronik

### Leistungsplatine

USB Spannung an Ladeelektronik, Daten an Microcontroller.

Der USB-Anschluss wird durch die Ladeelektronik mit maximal 1.27A belastet.

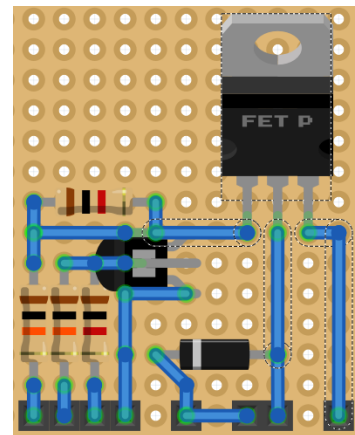
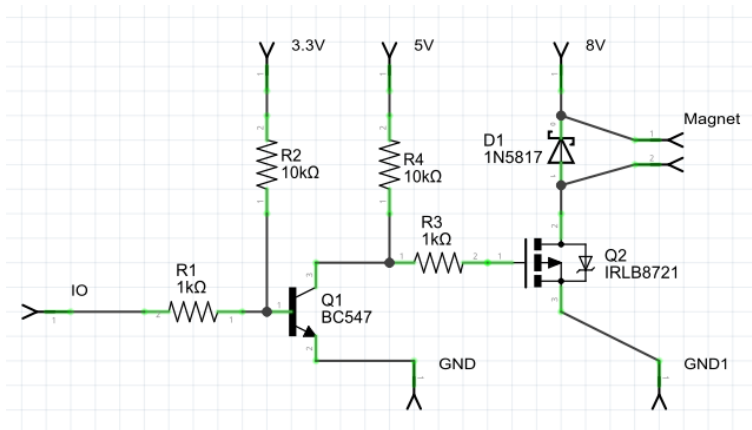
Die Ladeschlussspannung liegt bei 4,15V pro Akku.

Eine zusätzliche Schutzschaltung verhindert Über- und Tiefentladung.

Akkuspannung über Schalter an Kontrollplatine und Magnetschalter.

Magnetschalter erhält GND, 3.3V und 5V Spannung von Kontrollplatine.

### Entwurf Magnetschalter:



Durch den vorgeschalteten NPN Transistor kann der FET mit höherer Spannung durchgeschaltet werden. Die 3.3V vom IO-Pin des Microcontrollers reichen hierzu i.d.R. nicht aus.

Auch muss die Magnetspannung bei nicht initialisierten und daher hochohmigem IO-Pin sicher AUS sein. Signal ist daher invertiert, IO auf LOW schaltet ein.

### Kontrollplatine

Geschaltete Akkuspannung an Step-Up Wandler für Schrittmotoren

und an Step-Down Wandler für die 5V Versorgung von LED und Sensorboard.

Bei einem Linearregler wären die Verluste durch die doch höheren Ströme zu hoch.

Aus den 5V werden über einen Linearregler die 3.3V abgeleitet.

Da die 3.3V nur minimal belastet werden, sind die Verluste gering und kein Kühlkörper erforderlich.

Microcontroller ESP32 WROOM mit 18 IO-Pins und zwei AD-Pins.

Zwei Schrittmotortreiber für die Bewegungssteuerung.

## Anschlüsse und Komponenten der Grundplatte

### USB

D1 Uu (rot)    D2 D+ (grün)    D3 D- (weiss)    D4 GND (schwarz)

Die USB Spannung wird ausschließlich für das Laden verwendet.

Der zuerst gewählte Ansatz auch den Microcontroller mit zu versorgen bringt Probleme mit doppelter Spannungsversorgung und Rückspeiseeffekten aber keine nutzbaren Vorteile.

### Ein/Aus-Schalter

Schaltet den Akku auf die Leistungsplatine.

Die selbstabschaltende USB-Ladeelektronik ist permanent mit Akku und USB-Anschluss verbunden.

### Buzzer

Piezosummer für einfache Tonsignale.

Direkt vom Mikroprozessor über einen I/O angesprochen.

### Sub-D Steckverbindung Grundplatte / Korpus

Aus platztechnischen Gründen sitzen die Sensoren-Multiplexer am Deckel rechts, über dem Akku.

Die zentrale Steckverbindungen liegt vorne, sonst besteht Gefahr durch den fahrenden Magneten.

- 1-8    Multiplexer Chip-Selects und Adressen (braun) [CS1, CS2, CS3, CS4, A0, A1, A2, A3]
- 9      Analogsignal des gewählten Sensors (grün)
- 10    Data Neopixel LED (blau)
- 11    frei
- 12    V5 (rot) für Akku- und Status-LED
- 13    V5 (orange) für Sensorboard
- 14    Korpus (gelb)
- 15    GND (schwarz)

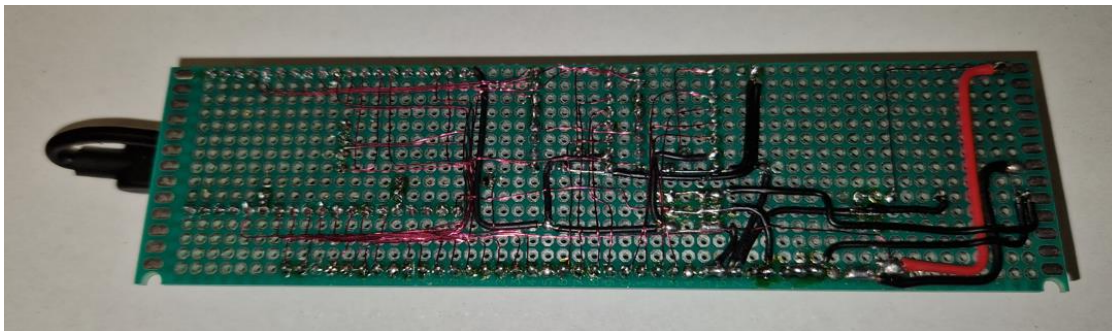
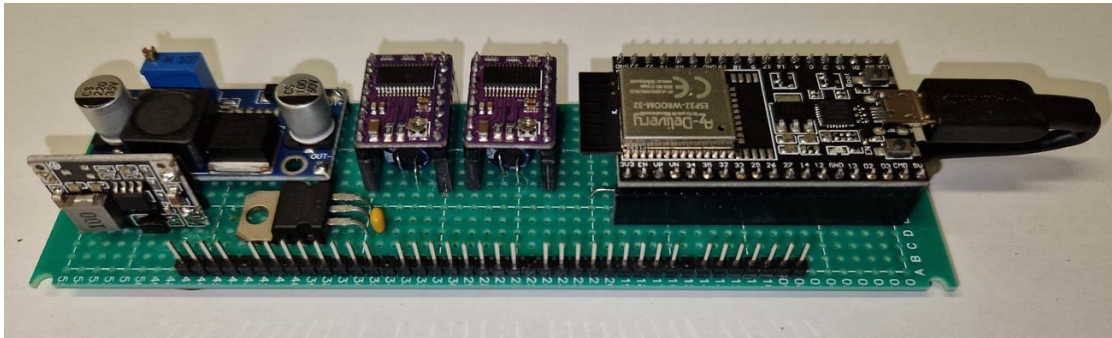
Lötbrücke in Korpus Buchse zwischen GND (15) und Korpus (14), somit kann der Microcontroller erkennen ob der Korpus auf der Grundplatte steht, also ob das Gehäuse geschlossen ist.

Die separate Versorgung des Sensorboards erlaubt ggfs. Spannungsanpassungen.



## Kontrollplatine

Maße: 150x40mm, 4mm Rand links und rechts, Bestückungsaufbau max. 25mm über Platine!



### Spannungsversorgung

Eingangsspannung vom Akku, also zwischen 7.2V und 8.4V.

Step-Up-Wandler auf die erforderlichen 9V für die Schrittmotortreiber einstellen.

(Es wäre auch möglich die Schrittmotoren direkt zu versorgen, da Eingangsspannung ausreicht.)

Der Step-Down-Wandler liefert 5V mit max. 1.5A für LEDs, Sensorboard und den 3.3V Linearregler.

Dieser produziert deutlich weniger Abwärme als ein Linearregler 7805.

LD33V 3.3V Linearregler, max. 0.8A, mit 10nF zusätzlich entstört, für Microcontroller.

Achtung, der LD33V (GOI) hat eine andere Pinbelegung als ein üblicher 78xx (IGO)!

Spannungsteiler 2200Ω/1000Ω, um Akkuspannung an den Messbereich des Microcontrollers anzupassen. Entsprechende Umrechnung in der Software erforderlich.

### Schrittmotor-Module



Die zwei Module auf Basis des DRV8825 werden mit 9V vom Step-Up-Wandler versorgt. Unter den Modulen ist jeweils der erforderliche Pufferkondensator mit 100μF verbaut.

Als Nennstrom sind lediglich 350mA eingestellt, dies entspricht einer Spannung am Poti von 175mV ( $U=0.5\Omega \cdot I$ ).

Hierdurch ist kein Kühlkörper erforderlich.

Das Drehmoment ist mehr als ausreichend.

Die Module laufen im geräuscharmen 1/16-Mikroschritt Modus (6400 Steps/Umdrehung).

Auflösung der Mechanik entspricht umgerechnet 80 Steps/mm.

Minimale Beschaltung mit Reset, Sleep und M2 an V3.3.

Step, Dir und Ena werden jeweils über separate I/O-Pins gesteuert.

## Microcontroller

### AZDelivery ESP32 Dev Kit C V4

512KB RAM, 2MB Flash, 80-240 MHz Taktfrequenz, 4 AD-Wandler, 19 digital I/O-Pins, USB-Interface.

Optimal für den Einsatz von Micropython mit Thonny oder C++ mit Arduino.



Beschaltung:

Nur Input:

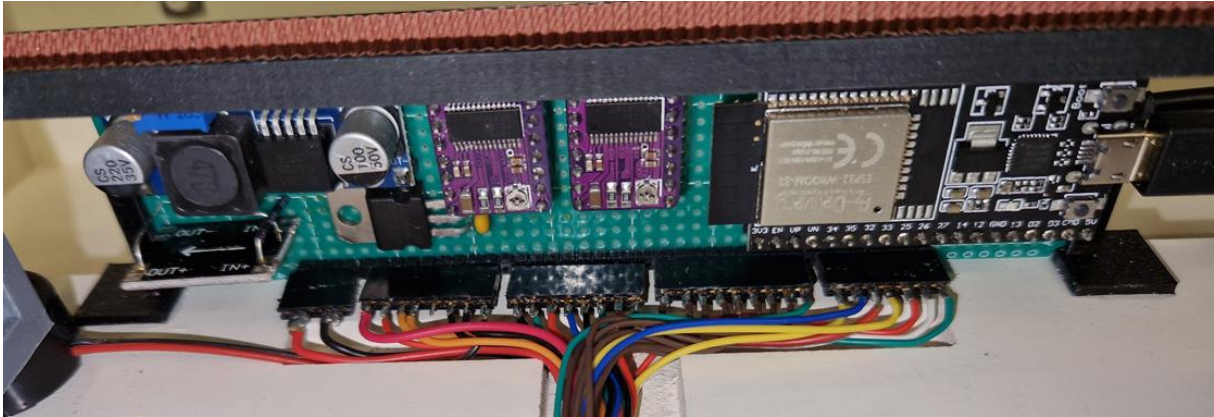
- 34 Akkuspannung (ADC1\_6)
- 35 Multiplex Data (ADC1\_7)
- 36 frei (ADC1\_0)
- 39 frei (ADC1\_3)

Generell I/O:

- |    |                  |               |              |
|----|------------------|---------------|--------------|
| 32 | Multiplex 0      | Chip-Select 1 | (ADC1_4)     |
| 33 | Multiplex 1      | Chip-Select 2 | (ADC1_5)     |
| 25 | Multiplex 2      | Chip-Select 3 |              |
| 26 | Multiplex 3      | Chip-Select 4 |              |
| 27 | Multiplex 4      | Adresse 0     |              |
| 14 | Multiplex 5      | Adresse 1     |              |
| 12 | Multiplex 6      | Adresse 2     |              |
| 13 | Multiplex 7      | Adresse 3     |              |
| 23 | Stepper Enable A |               | (VSPI-MOSI)  |
| 19 | Stepper Enable B |               | (VSPI-MISO)  |
| 15 | Stepper Step A   |               |              |
| 17 | Stepper Step B   |               |              |
| 02 | Stepper Dir A    |               |              |
| 16 | Stepper Dir B    |               |              |
| 18 | FET-Magnet       |               | (VSPI-Clock) |
| 05 | LED NeoPixel     |               | (VSPI-SS)    |
| 22 | Beeper           |               | (I2C-CL)     |
| 21 | Korpus           |               | (I2C-DA)     |
| 04 | Reserve          |               |              |

Zusätzlich D+ und D- der USB Schnittstelle für die Programmierung.

Steckerleiste 40polig (Aufteilung 4 8 8 9 7)



Versorgungsspannung, Einspeisung aus zwei 4.2V LiPo Zellen, also 7.2V bis 8.4V.

01-02 2 \* Ub geschaltete Akkuspannung, doppelt wegen Strombelastung

03-04 2 \* GND GND, doppelt wegen Strombelastung

Ausgangsspannungen

06-08 V5 Korpus LED, Leistungsplatine, Korpus Sensor

09-10 V33 Leistungsplatine für FET-Schutz, frei

11-13 GND Korpus, Leistungsplatine, Beeper

Schrittmotorenanschlüsse

15-18 Stepper 1 (W1: 15+17, W2: 16+18)

19-22 Stepper 2 (W1: 19+21, W2: 20+22)

Sensorboard

24-27 Mpx Chip-Select

28-31 Mpx Adresse

32 Mpx Data (Eingang AD, Analogspannung)

I/Os

34 Magnet Ausgang Magnet an/aus

35 Neopixel Data Ausgang Neopixel Datentelegramme

36 Beeper Ausgangsspannung Tonsignalgeber

37 Korpus Eingang für Erkennung Korpus offen/geschlossen

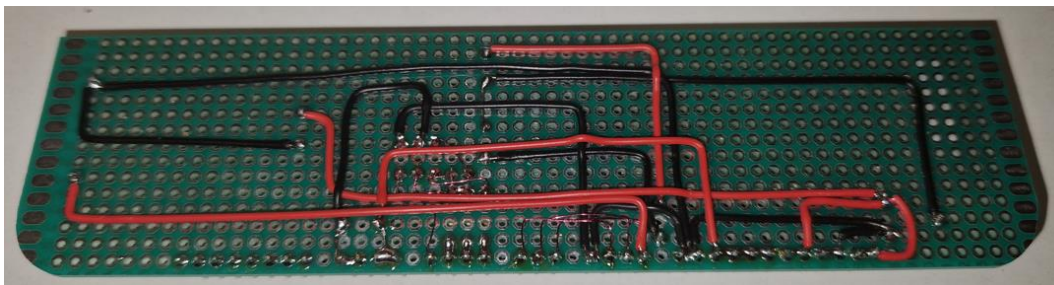
38 Uu USB Spannung, z.B. für Microcontroller, aktuell nicht genutzt

39 D- USB D- für Microcontroller (weiß)

40 D+ USB D+ für Microcontroller (grün)

## Leistungsplatine

Maße: 150x40mm, 4mm Rand links und rechts, Bestückung max. bis 25mm über Platine!



### USB-Ladeelektronik / Akku

Die Spannung am USB-Eingang dient ausschließlich dem Laden des Akkus.

Die Datenleitungen des USB-Anschlusses werden an den Microcontroller weitergegeben.

Spezialmodul zum Laden von zwei LiPo Akkus und die Schutzschaltung verhindert Über- und Unterspannung der einzelnen Zellen. Die Akkus werden jedoch nicht ausbalanciert.

Der Akku ermöglicht kurzfristig die zehnfache Abgabeleistung als der reine USB-Anschluss.

Die Leistungsaufnahme, also ohne gerade fahrende Figuren, beträgt lediglich 2.5 Watt.

Somit wird der USB-Anschluss zumeist mit nur 0.5A belastet.

Die maximale Belastung des 5V USB-Anschlusses beträgt beim Laden 1.27A also 6,35W.

Der maximale Ausgangsstrom kann bei ca. 8V kurzzeitig bis 8A, also 64W, betragen.

Real werden aber nur Leistungsspitzen im Bereich um 10-20 Watt erreicht.

Die Akkuspannung wird über den Ein-/Ausschalter an die Magnetsteuerung und die Kontrollplatine geführt. Die Schutzschaltung und die Ladeelektronik ist immer mit dem Akku verbunden.

Es ist erwartungsgemäß keine Entladung im ausgeschalteten Zustand zu beobachten.

### Magnetsteuerung

N-Kanal FET IRLB8721 mit NPN Transistor BC547 zur Erhöhung der Gate-Spannung und definierter Abschaltung bei nicht aktivem, also schwebendem, I/O-Pin.

Eingang mittels 10K Widerstand auf 3.3V und Ausgang mittels 10K Widerstand auf 5V ziehen.

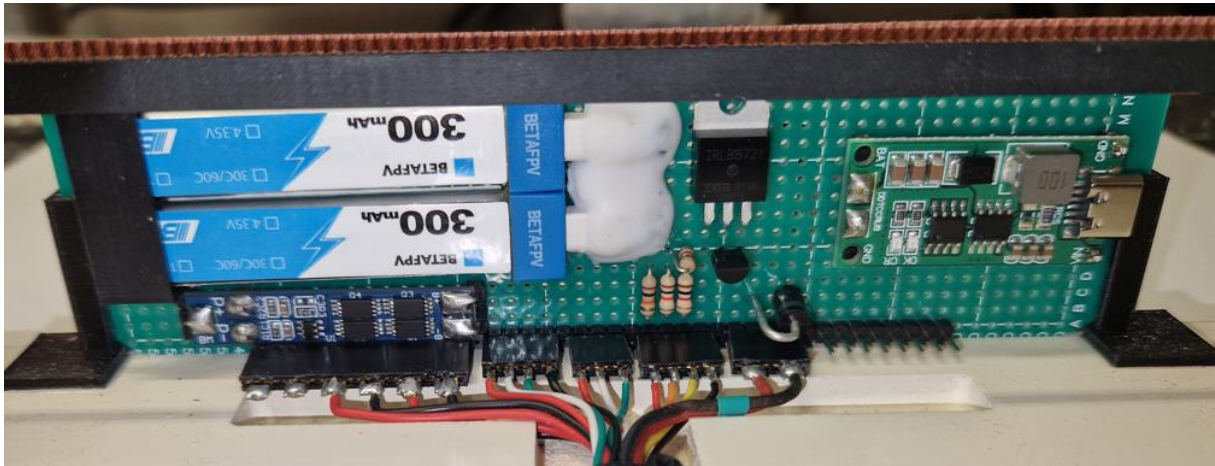
Zusätzlich sind Ein- und Ausgang mit 1K Widerständen gegen Rückkopplungen geschützt.

Funktion ist invertiert, d.h. der Magnet wird bei Ausgabe einer logischen 0 eingeschaltet!

### Magnetspannung

Die Versorgungsspannung liegt mit 7.2V bis 8.4V deutlich über der Betriebsspannung von 5V des Elektromagneten. Da dieser nur in kurzen Intervallen eingeschaltet wird ist diese Überbelastung ok.

Steckerleiste 40polig (12 4 3 4 4 8)



Akkuanschluss 4.2V LiPo Zelle, Schalter und Ausgangsspannung für Kontrollplatine.

|       |             |  |
|-------|-------------|--|
| 01-02 | 2 * Ua+     | Reserveanschluss Akku                  |
| 03-04 | 2 * Ua-     |  |
| 05-06 | 2 * Sw      | Schalter, doppelt wegen Strombelastung |
| 07-08 | 2 * Sw      |  |
| 09-10 | 2 * Ua+     | Ausgang, doppelt wegen Strombelastung  |
| 11-12 | 2 * Ua-/GND | (GND: zentraler Massepunkt)            |

USB-Anschlussblende

|       |               |                          |
|-------|---------------|--------------------------|
| 14-17 | Uu, D- D+ GND | rot, weiß, grün, schwarz |
|-------|---------------|--------------------------|

USB Spannungs- und Datenweitergabe an Kontrollplatine

|       |           |                 |
|-------|-----------|-----------------|
| 19-21 | Uu, D- D+ | rot, weiß, grün |
|-------|-----------|-----------------|

Eingänge für Magnetkontrolle durch Kontrollplatine

|       |                 |                                   |
|-------|-----------------|-----------------------------------|
| 23-26 | V5, V33, M, GND | 5V, 3.3V, Logiksignal Magnet, GND |
|-------|-----------------|-----------------------------------|

Magnetausgang

|       |        |                              |
|-------|--------|------------------------------|
| 28-31 | Magnet | doppelt wegen Strombelastung |
|-------|--------|------------------------------|

Reserve

|       |      |  |
|-------|------|--|
| 33-40 | frei |  |
|-------|------|--|

## Korpus

Kasten mit Innenmaßen von 292x55mm, unten offen.

Bei 4mm Wandstärke wird das dann eine Kiste mit 30x30x8 cm.

Grundplatte mit Mechanik wird wie beschrieben verdeckt von unten in einen Rahmen eingeschraubt.

Die 1mm starke Abdeckung, auf die das Schachbrett gedruckt/gelasert wird, kommt später.

So hat man bei der Montage noch freie Sicht nach innen und die Optik und Maßhaltigkeit des 1mm starken Furnier Deckholzes ist wesentlich besser als bei einer ausgefrästen Multiplex Deckplatte.

Alle elektrischen Verbindungen von der Grundplatte an den Korpus werden beim Schließen des Gehäuses über die fest eingebaute Sub-D Steckverbindung automatisch hergestellt.

Über den unten und oben offenen Kasten kommt das mit dem Design gelaserte Furnier.

Zur besseren Positionierung des Layouts wird das Furnier erst verklebt und anschließend gelasert.

Als Verstärkung ist das Furnier im Bereich des Ausschnittes mit einer 1mm GFK-Platte verstärkt.

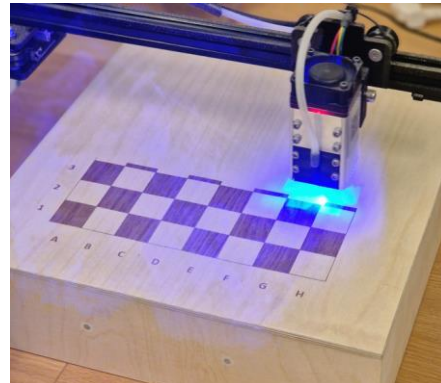
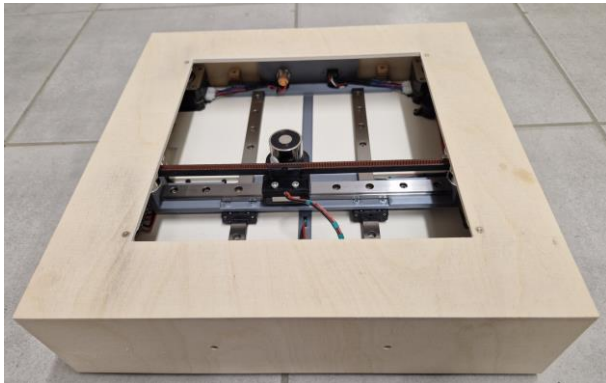
Das 3mm dicke Sensorboard wird von unten komplett im Deckel, bis auf die GFK-Platte, versenkt

Der Magnet muss möglichst nahe an der Deckeloberfläche fahren können.

Die Abstandhalter mit der Gehäuseverschraubung haben unterschiedliche Höhenlagen.

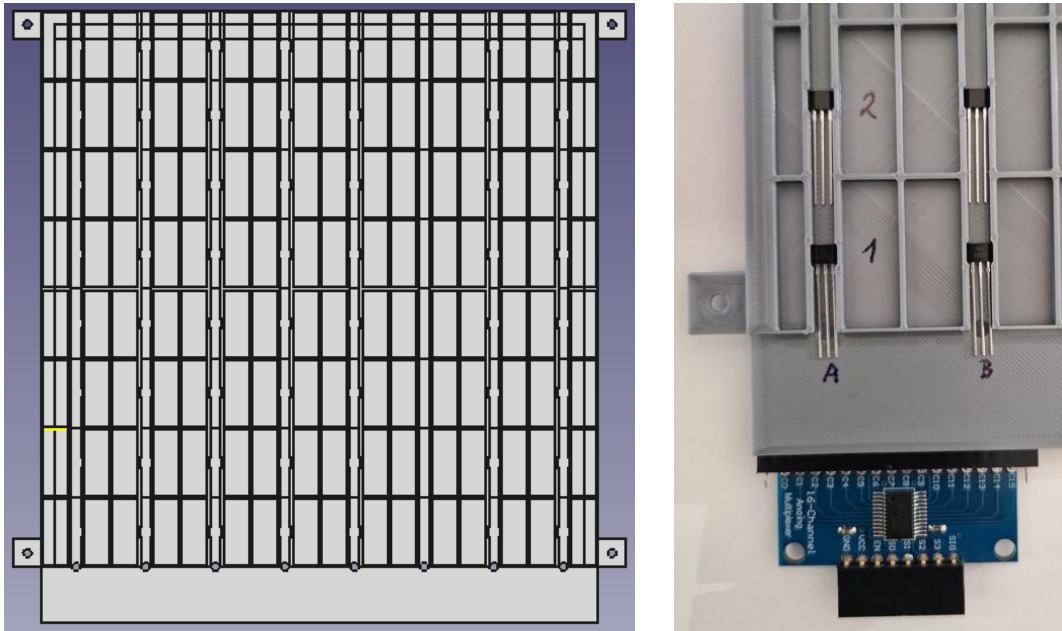
Diese ist vorne 54mm und hinten, wegen der 5mm starken Anschlussblende, 49mm.

Jeweils von der Auflagefläche zur Unterseite Deckel gemessen.



## Sensorboard

Das Sensorboard ist 3mm dick. Es wird 2mm in die Deckplatte eingelassen.



Ansicht von oben. Beispiel Multiplexer 1 mit den HAL Sensoren für Felder A1, A2, B1 und B2.

Jedes Spielfeld erhält einen HAL Sensor vom Typ 49E.

Die Sensoren mit der Beschriftung nach oben, also zum Magneten der Figur, einbauen.

Die Ausgangsspannung des Sensors liegt z.B. bei Betriebsspannung von 3.3V im Bereich 1V bis 1.6V.

Um Abwärme des 3.3V Linearreglers zu reduzieren mit 5V Betriebsspannung versorgt.

Bei falscher Magnetfeldrichtung besteht bei 5V theoretische Gefahr der Überspannung am AD-Eingang des Microcontrollers, aber nur bei sehr starken Magneten.

64 Felder in 4 Gruppen a 2\*8 Felden. Jede Gruppe an einem Multiplexer 78HC4067.

Die komplette Sensorik benötigt knapp 300mA.

Auf den ersten Blick Stromsparpotential durch geschaltete Spannungsversorgung um die Sensoren nur bei Bedarf zu versorgen. Praktisch wird das System die meiste Zeit das Sensorboard abschannen um Aktionen des Spielers zu erfassen. Daher nicht weiterverfolgt.

Das Sensorboard hat unten auf der Hinterseite für vier Multiplexer Steckerleisten mit je 18 Pins.

Das Anschlusschema (von oben (Spielseite) gesehen) ist:

Multiplexer 1: GND A1 A2 A3 A4 A5 A6 A7 A8 B8 B7 B6 B5 B4 B3 B2 B1 Vcc

Multiplexer 2: GND C1 C2 C3 C4 C5 C6 C7 C8 D8 D7 D6 D5 D4 D3 D2 D1 Vcc

Multiplexer 3: GND E1 E2 E3 E4 E5 E6 E7 E8 F8 F7 F6 F5 F4 F3 F2 F1 Vcc

Multiplexer 4: GND G1 G2 G3 G4 G5 G6 G7 G8 H8 H6 H5 H7 H4 H3 H2 H1 Vcc

(Fehler in der Reihenfolge lassen sich in der Software leicht beheben.)

## NeoPixel LEDs

Status LEDs, welche einen überstrichenen Bereich der Frontplatte beleuchten.

Bei Versorgung des Neopixel mit 5V ist ein 10K Vorwiderstand vor Data erforderlich, da der Microcontroller nur 3.3V ausgibt, der Neopixel aber eine Schaltschwelle von  $0,7 \cdot V_{cc}$  also 3.5V hat.

Durch den Vorwiderstand wirkt der interne Pullup des Neopixel zuverlässig.

Farben der Anschlüsse: rot Vin 5V, gelb Din, grün Dout-Din (2. Led), schw. GND

3 polige Steckverbindung Data/V/GND

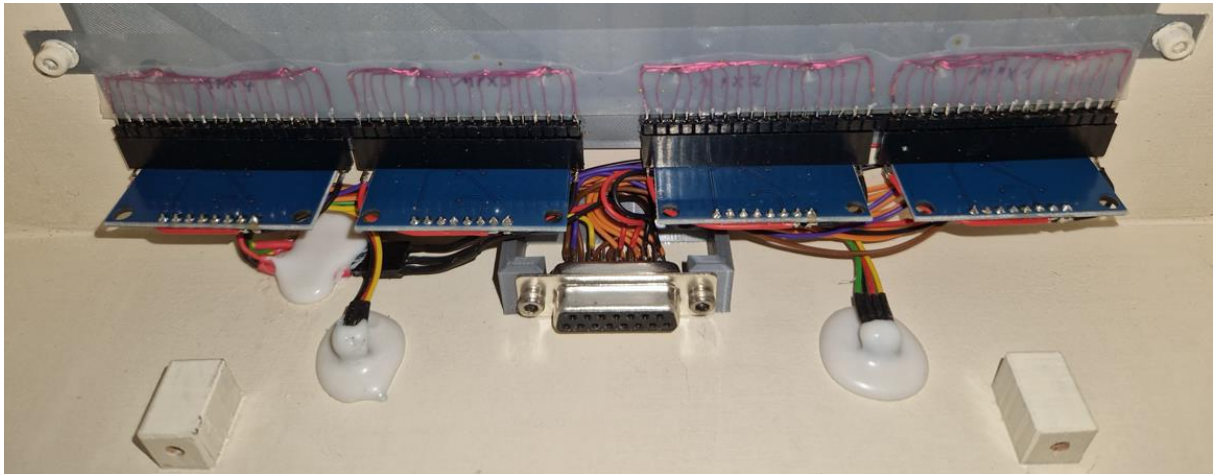
## Einbau Elektronik

Die Elektronik muss in den vorderen oberen Bereich vor das Sensorboard und über die Steckverbindung. Ansonsten kollidiert der Magnetschlitten. Eng, aber ausreichend.

Die feinen Anschlüsse des Sensorboards wurden zum Schutz vor dem Magnet später mit festem Isolierband geschützt. Ebenso der Übergang Korpus/Sensorboard auf der gegenüberliegenden Seite.

Die vier Multiplex-Platinen sind bereits aufgesteckt und mit dem Sub-D Steckverbinder verbunden.

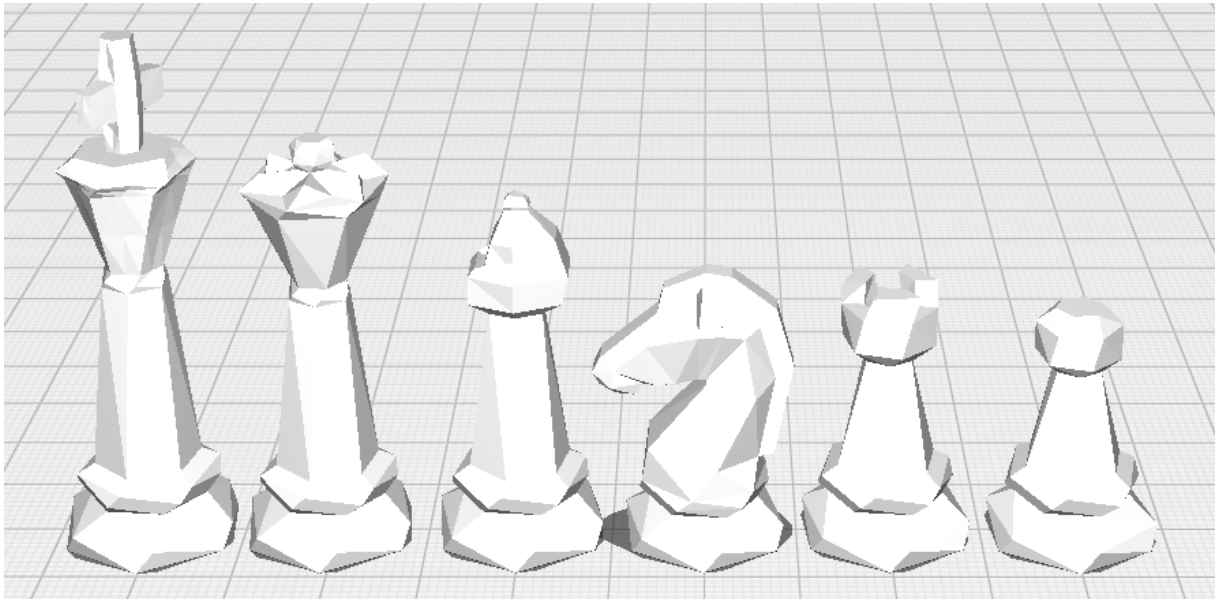
Unten sind die Auflagen für die Grundplatte mit Einschlagmuttern zur Verschraubung zu sehen.



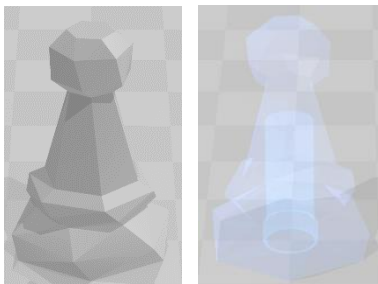


## Schachfiguren

Die Figuren sind aus dem Low-Poly Chess-Set von Matt Kerekanich von Thingiverse abgeleitet. Figurendurchmesser knapp 20mm, Gewicht zwischen 4,1 und 6,5 Gramm.



Gedruckt wird mit schwarzem und weißem PLA bei 0.12mm Schichthöhe.



Neben der Größenanpassung ist eine Vertiefung für den 6x2mm Neodyn-Magneten und ein Hohlraum für das Messinggewicht eingearbeitet.

Als Gewicht dienen 20mm lange Abschnitte eines 5mm Messing-Rundstabes. Bei dem Springer ist nur Platz für 19mm.

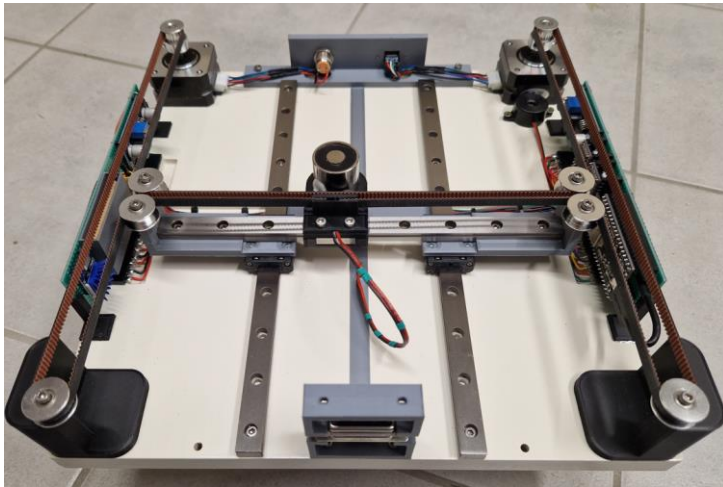
Die Gewichte wiegen nur rund 2g, sind aber deutlich spürbar. Auf die Ausrichtung des Magneten ist natürlich zu achten!



Nach dem Druck wird das Gewicht und der Magnet mit Epoxidharz in die Figur eingeklebt. Ein 0.3mm dünner Filz unter den Figuren schützt und gibt ein besseres Stell- und Gleitgefühl.

## Detailaufnahmen

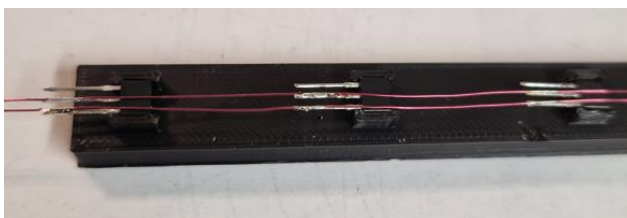
Mechanik:



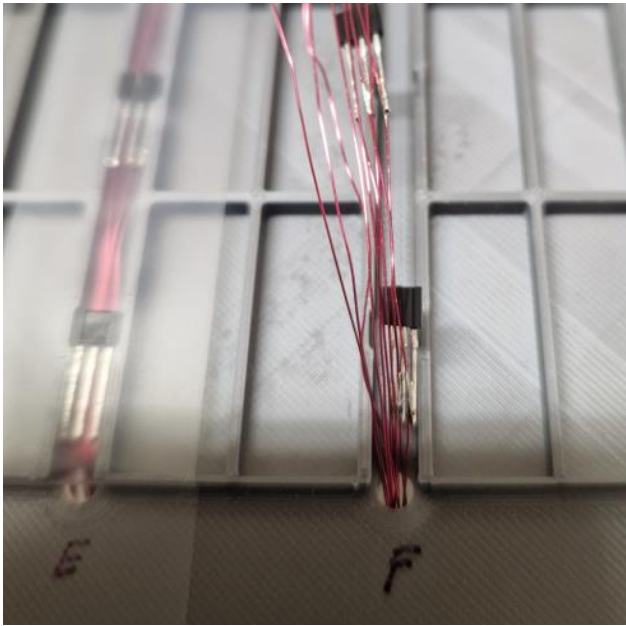
Testkabel um auch im offenen Zustand alles Messen zu können.



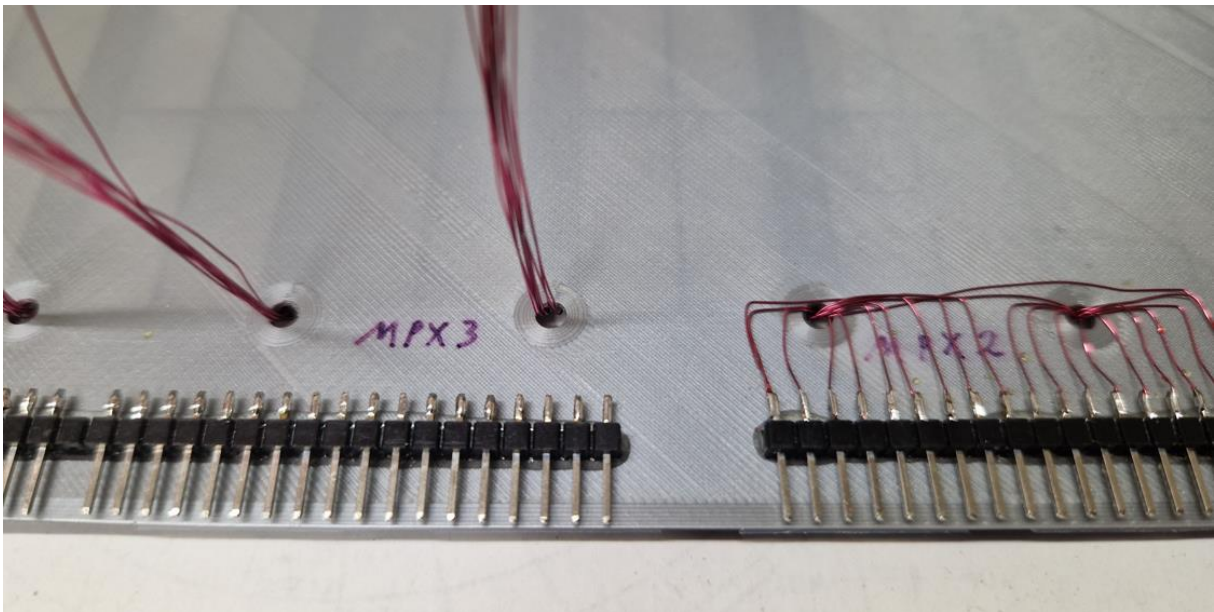
Um die HAL-Sensorstränge mit den korrekten Abständen herzustellen, wird eine einfache Löthilfe verwendet.



Einbau der Sensoren in das Sensorboard.

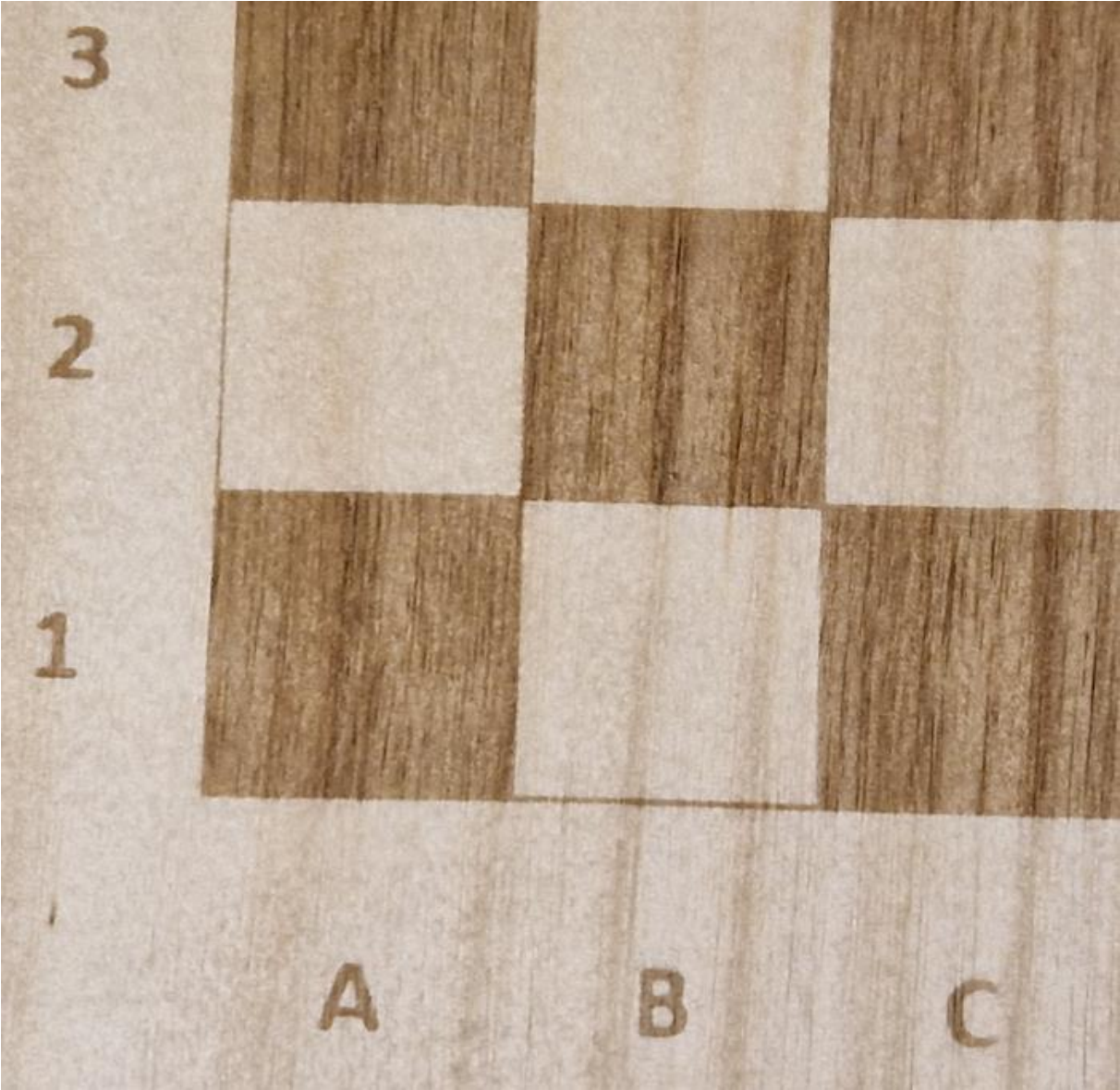


Steckverbindungen der Multiplexer pro HAL-Sensorstrang herstellen



Dies ist die Unterseite, also ist GND außen rechts und Vcc außen links.  
Die Sensorsignale werden daher von rechts nach links aufgelegt.

Gelasertes Layout im Detail



## Farben & Design

|             |          |           |  |
|-------------|----------|-----------|--|
| Grundplatte | RAL 9001 | cremeweiß | Kunstharz (Bauhaus, swingcolor Buntlack) |
| Korpus      | RAL 9001 | cremeweiß | Kunstharz (Bauhaus, swingcolor Buntlack) |

zusätzlich zur Versiegelung:

|           |          |            |  |
|-----------|----------|------------|--|
| Oberseite | Klarlack | seidenmatt | Kunstharz (Bauhaus, swingcolor Buntlack) |
|-----------|----------|------------|--|

Lasergravur der mit Klarlack vorgestrichenen Oberseite mit 5000 mm/min mit 66% Leistung (entspricht ca. 25W), danach mehrfacher Klarlackauftrag.

## Verlauf

|            |   |
|------------|---|
| 02.04.2022 | Ideensammlung   |
| 10.04.2022 | Grundplatte mit XY-Mechanik aufgebaut   |
| 24.04.2022 | Elektronik auf der Grundplatte verbaut  |
| 18.05.2022 | Basisversion der Schach-Engine Version 0.24 bereit                                |
| 20.05.2022 | Grundbrett Energieversorgung verkabelt und getestet                               |
| 01.06.2022 | Korpus gebaut und Lackierung begonnen   |
| 05.06.2022 | Layout gelasert, Leistung Magnet ungenügend, Figur folgt nicht immer              |
| 05.02.2023 | Neuanfang, Design mit 2 Einzelplatinen und Leitungskanälen                        |
| 21.02.2023 | Kontrollplatine entworfen und Grundplatte gefertigt                               |
| 22.03.2023 | Fertigstellung Grundplatte inkl. Mechanik, Kontrollplatine, Leistungsplatine usw. |
| 02.04.2023 | Erste Spalte der Sensorplatte aufgebaut und getestet                              |
| 17.04.2023 | Sensorplatte funktioniert, sehr gute Selektivität und Empfindlichkeit             |
| 18.04.2023 | Stromverbrauch Sensorplatte durch Wandler >1A, direkt vom Akku versorgen          |
| 19.06.2023 | Mechanische Dauertests, Stromaufnahme 650mA in Ruhe                               |
| 24.06.2023 | Austausch 5V Linearregler gegen 5V Schaltregler, auch für die HAL Sensoren        |
| 07.07.2023 | Neue Leistungsplatine mit 2 LiPo Akkus aufgebaut, Magnet jetzt stark genug        |
| 24.07.2023 | Neudesign der Schachfiguren um das Gewicht über 4g zu erhöhen                     |
| 08.08.2023 | Abschluss des Projektes   |

## Resumee

Der Ansatz mit Micropython hat sich bewährt. Ausreichend schnell und stabil in der Entwicklung. Software und der mechanische Aufbau waren wie erwartet komplex, aber lösbar.

Der hartnäckige Versuch mit einer Akkuzelle auszukommen hat viel Zeit gekostet und wurde letztendlich verworfen.

Das Ziel ausschließlich mit einem USB-Anschluss zu arbeiten konnte trotzdem erreicht werden.

Jetzt kann weiter an der Spielstärke gearbeitet werden ...