

# Bauanleitung Mini-Remote-Sternwarte für das Seestar S50



## Inhalt:

- 1) Disclaimer
- 2) Problemstellungen und Anforderungen
- 3) Materialliste und Kosten
- 4) Schutzbau und Mechanik
- 5) Technik für die Steuerung
  - 5a) Der Anschalter für das Seestar
  - 5b) Kabelzuführung bei Drehung  $>360^\circ$  und Stativ-Modifikation
- 6) Steuer-Elektronik
  - 6a) Software
  - 6b) Steuerungs-Hardware und die Verkabelung der Komponenten
- 7) Fernzugriff mit scrcpy



## **1) Disclaimer**

Diese Anleitung dient lediglich der Dokumentation oder als Anregung für eigene Projekte. Nachbau ist ausdrücklich erlaubt, jedoch hat jeder selbst Sorge für Sicherheit im Rahmen der geltenden Vorschriften und VDE-Richtlinien – insbesondere bei der Montage von Steckdosen, der Blitzschutzterdung, des Brandschutzes und der Sturmsicherung zu tragen. Der Autor übernimmt hierfür keinerlei Verantwortung.

Irrtümer und fehlerhafte Teile innerhalb dieser Anleitung sind nicht auszuschließen, im Zweifel bitte ich um Kontakt unter [sternwarte@muchelndorf.de](mailto:sternwarte@muchelndorf.de).

Die Verwendung des modifizierten LineageOS erlaubt lediglich eine nicht kommerzielle Nutzung!

## **2) Problemstellungen und Anforderungen**

- Das Seestar soll in einem wettergeschütztem Gehäuse komplett fernbedienbar sein.
- Das Seestar lässt sich nur über Android oder IOS ansteuern
- Das Seestar braucht eine Bluetooth-Verbindung zur Legitimation der WLAN-Verbindung
- Das Seestar soll permanent mit Spannung versorgt werden
- Das Seestar kann >360° drehen, die Zuführungskabel müssen mitdrehen
- Das Seestar soll aus der Ferne eingeschaltet werden können

### **3) Materialliste und Kosten**

#### **Gehäuse:**

Kiste: 100€

Alu Vierkantrrohr 20x20x2, VA-Schrauben, Muttern, Unterlegscheiben: 50€

Gewindestange M10, Alurohr, M19 → 3/8“ Adapter: 20€

Alu Winkel 100x100 für Deckel: 40€

PVC-Platte 1000x500x3: 20€

Montagekleber Alu auf PVC: 10€

Dichtmittel | Silikon | Acryl | Metall-Knete: 30€

Gummidichtung 1 | 2 = 26€

Kleinkram: 10€

**Gesamt: 306€**

#### **Technik:**

Aktuator: 40€

Netzteil 12V/5A: 15€

Raspberry Pi4 8GB: 70€

Raspberry Netzteil: 10€

SD-Karte 64GB: 12€

Rasbpi-Gehäuse mit GPIO-Auslass: 13€

Motortreiber L298N: 6€

Hub/Zugmagnet zur Schalterbetätigung: 9€

Schleifring mit Achsdurchführung: 40€

Verteilerkästen (2x, verschiedene Größen): 50€

Fritzbox 7530AX: 150€

Webcam: 5€

Bluetooth-Dongle: 14€

Relais-Modul: 10€

Lüftersteuerung: 8€

Steckdosen 2er Feuchtraum, Gira 2x: 40€

**Gesamt: 452€ → Gesamtkosten ca. 800€**

#### **4) Schutzbau / Mechanik:**

Ich habe eine Weile im Internet nach einer Kiste mit umlaufender Kante gesucht, in die das Seestar reinpasst. Beim großen Fluß bin ich auf diese stabile 70l Alukiste gestoßen:

<https://www.amazon.de/dp/B0BFB9PP8W?th=1&tag=astrotrastron-21>



Zusätzlich habe ich mir noch etwas Alu-Vierkant-Rohr (20x20x2mm) besorgt.

Nach dem Motto "keep it simple" habe ich mir einen einfachen Mechanismus konstruiert, der den Deckel komplett vom Kistenrand wegklappt. Ich habe den Deckel absichtlich höher angebracht, damit das Teleskop ungestört über den Rand blicken kann.

Der Deckel bekommt noch eine breite umlaufende Kante, die über das Kistenteilunterteil überlappt.

Zuerst habe ich alle Scharniere und Beschläge entfernt (Nieten aufgebohrt) und dann habe ich einen ganz einfachen Mechanismus konstruiert:



Der Linearmotor wird so an der Kiste montiert, dass er das Dach ganz aufklappen kann:



Die Aufhängung des Aktuator-Achse habe ich von innen etwas verstärkt – das recht dünne Aluminium der Kistenwand ist nicht stabil genug:



Der Deckel sollte eine möglichst breite, umlaufende Kante haben und möglichst leicht sein.

Ich habe mir eine [3mm PVC-Kunststoffplatte](#) zurechtgeschnitten und mit einem [Montagekleber für Aluminium und PVC](#) umlaufend [Aluwinkel \(100x100x0,8\)](#) verklebt.

Die Unterseite habe ich absilikoniert, die Winkel sind umlaufend mit 8mm Abstand zur PVC-Platte montiert, da soll dann auch noch eine Acryl-Fuge zum abdichten dran:



Damit der Regen besser ablaufen kann, habe ich das Dach abschüssig gestaltet.

Dazu habe ich zwei Reste Alu-Vierkantrohr zusammengeschraubt und an einer Seite des Kistendeckels befestigt:





Die Kiste hat innen noch einen Gummi-gepufferten Gehäuselüfter spendiert bekommen. Dafür habe ich ein Loch in die Kiste gesägt und von außen mit einer [Schutzabdeckung mit Insektenschutz](#) versehen:



Um die Kiste später nivellieren zu können (z.B. auf einer schrägen Dachfläche) habe ich noch ein paar verstellbare Gerätefüsse montiert:



Das Gehäuse hat eine Grundierung und ein mattschwarzes Finish bekommen. Alu ist zwar schick, aber so ein glänzendes Ding auf dem Schuppendach wollte ich irgendwie nicht.



Außerdem habe ich für den Wetterschutz noch ein paar Gummilippen montiert:



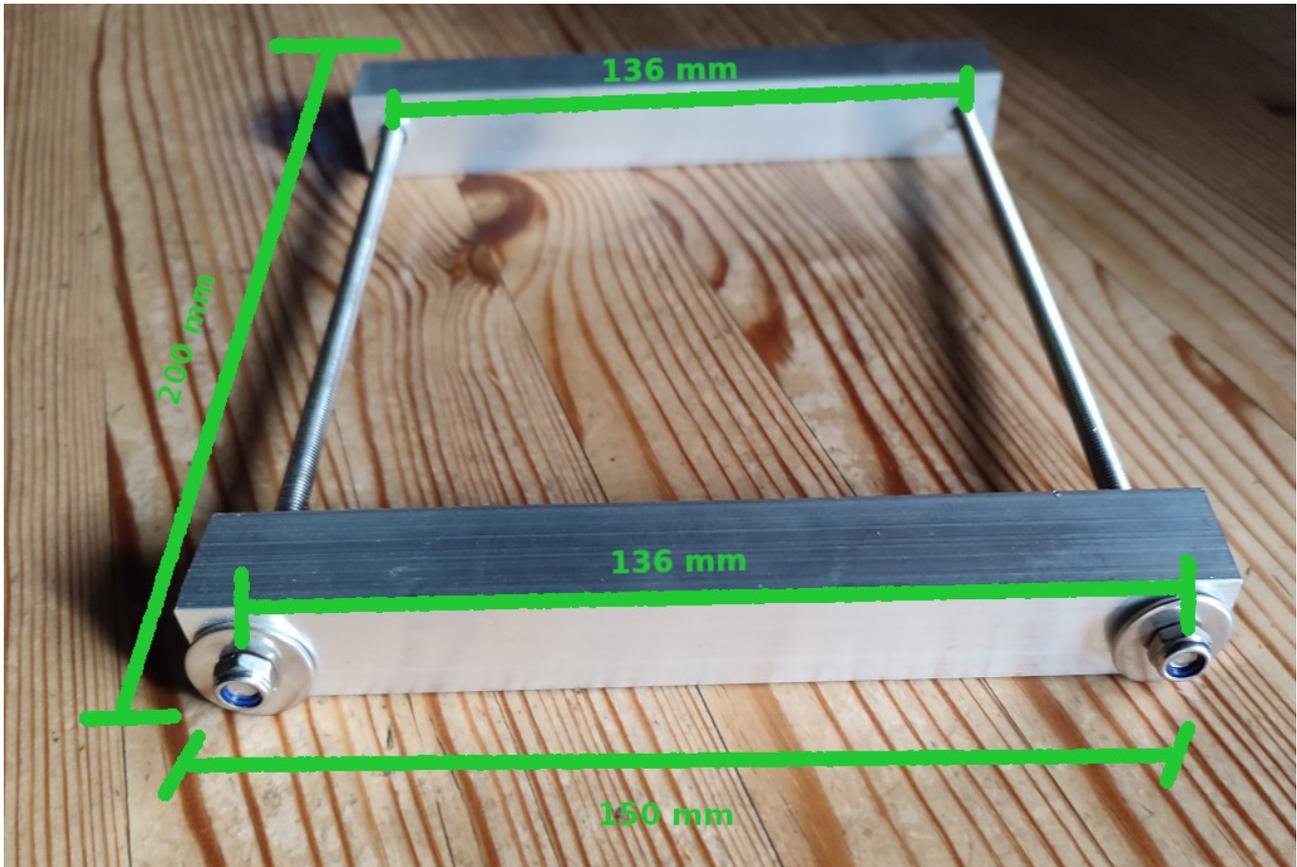
Der Aktuator ist eigentlich wetterfest und von oben durch das Dach abgeschirmt. Ich habe aber trotzdem nochmal ein DN75-Rohr plus Verschluss-Stopfen drübergestülpt:



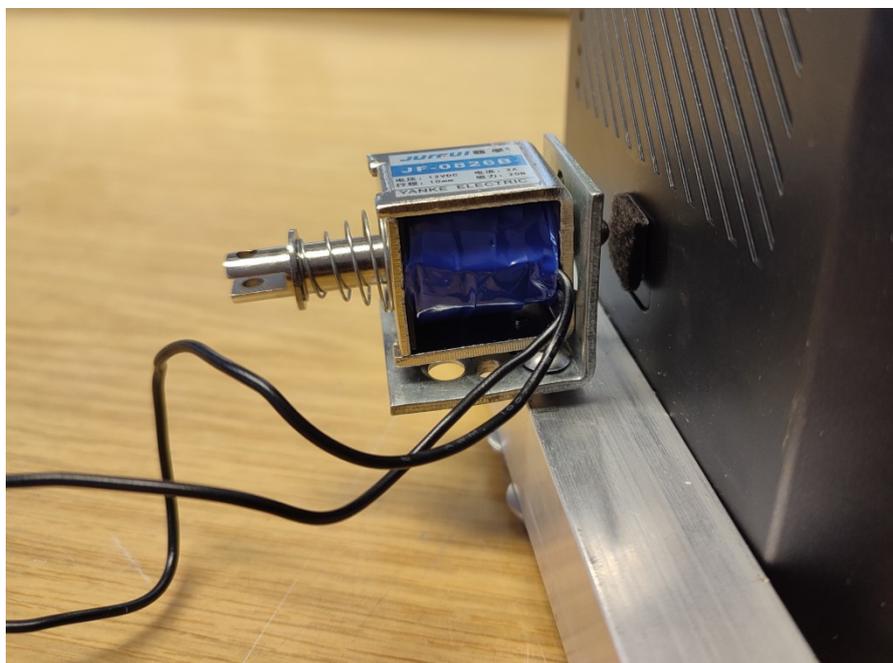
## 5) Die Technik für die Steuerung

### 5a) Der Anschalter für das Seestar

Ich wollte das Seestar-Gehäuse nicht öffnen, deshalb habe ich mir aus den Resten meiner Alu-4kant-Leisten und zwei M5-Gewindestangen einen Halter gebastelt:



An den Halter habe ich mit einem kleinen Winkel [einen 12V Hubmagneten](#) befestigt:

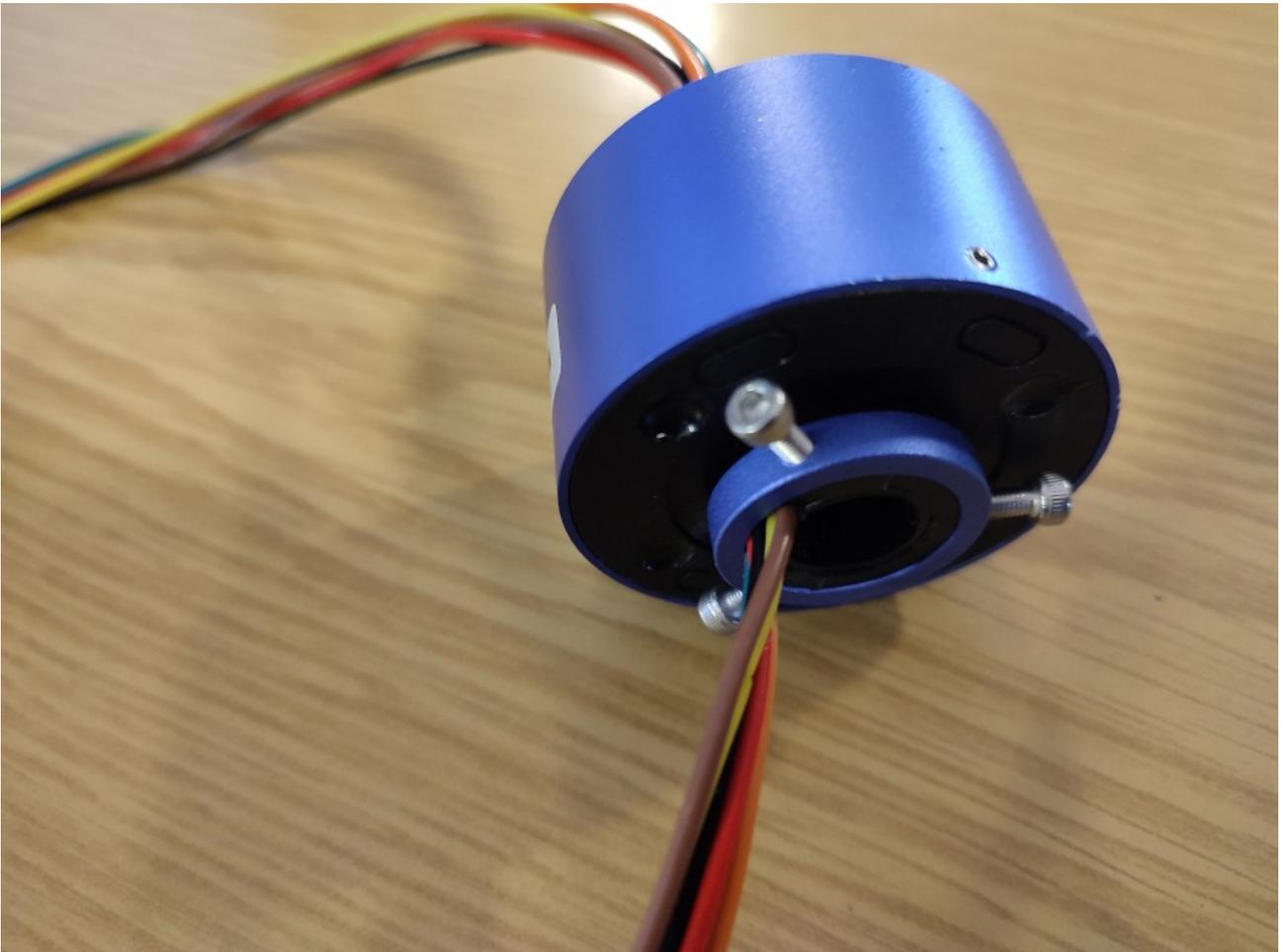


Weil das Teil mit 20N auf den Schalter drückt, habe ich zum Abdämpfen des "Impacts" ein Stück Möbelfilz auf den Knopf geklebt.

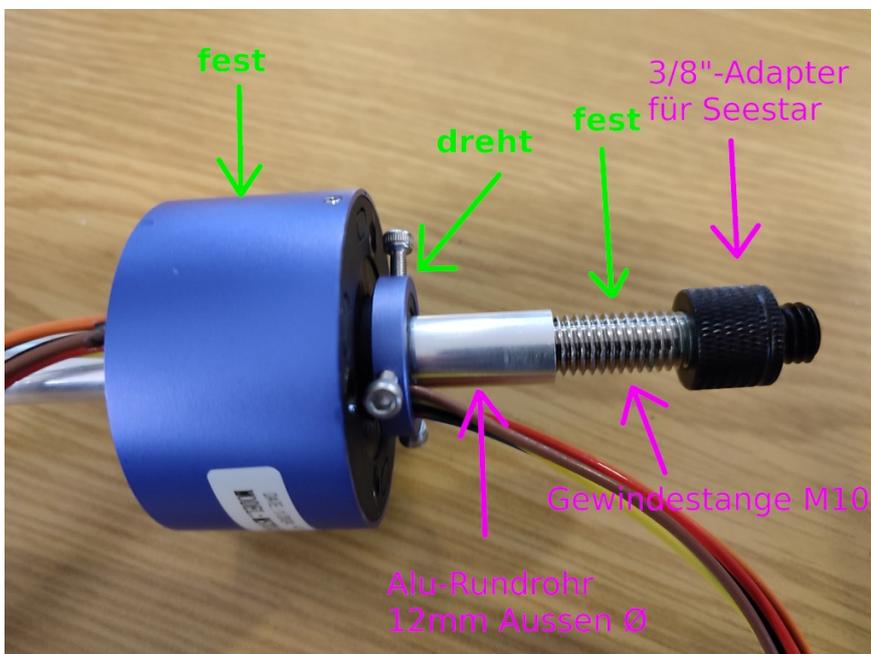


### 5b) Kabelzuführung bei Drehung >360° und Stativ-Modifikation:

Das Seestar hat die Eigenart, gelegentlich mehr als 360° zu drehen. Trotzdem soll es permanent mit Spannung versorgt werden. Ich habe mir dafür einen [Schleifring mit Achsdurchführung](#) besorgt:



Ich habe eine M10-Gewindestange in einem Alu-Rohr mit 12mm Aussendurchmesser durch den Schleifring gesteckt:



Zur Befestigung am Original-Stativ habe ich die 3/4" Schraube des Seestar-Stativs entfernt. Das war etwas krampfig - scheinbar ist die verpresst oder zumindest verklebt und ich musste die Schraube ausbohren. Dann habe ich das Loch passend für die M10-Gewindestange größer gebohrt und auf beiden Seiten des Stativ\_Tellers eine M10 Sicherungsmutter plus Scheibe aufgeschraubt. Die Muttern habe ich ordentlich angezogen, damit sich die Stange nicht mehr bewegt. Dann das Alurohr als Hülse und darauf den Schleifring:



Der Schleifring hat eine innen liegende Verdrehsicherung (= ein kleiner Flansch), die mit zwei Schrauben befestigt ist.

Die Sicherung habe ich abgeschraubt, so gedreht, dass sie nach aussen zeigt und mit einer zusätzlichen Schraube am Stativ gesichert.

Also: Gewindestange fest, Schleifring-Basis fest. Für die "nicht mitdrehenden" Kabel habe ich ein weiteres Loch in den Stativteller gebohrt.

Obendrauf sitzt der 3/4" -Adapter für das Seestar:



Damit es von der Höhe her passt, habe ich die Beine des Steestar-Stativs abgeflex.

Für eine Nivellierbasis ist in der Höhe leider kein Platz. Zwar hat die Kiste höhenverstellbare [Maschinenfüsse](#) bekommen, ich wollte mir aber trotzdem

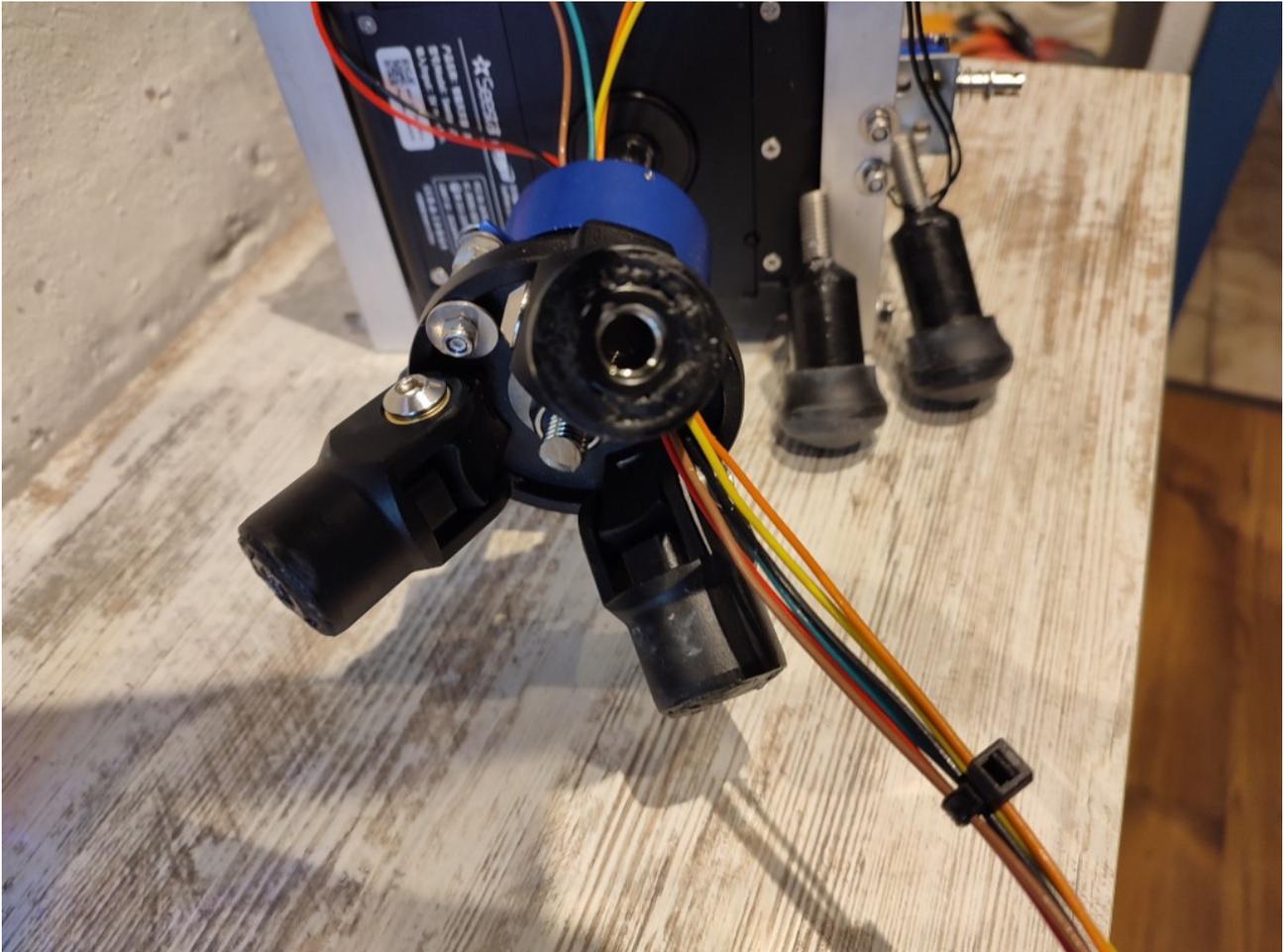
die Verstelloption am Seestar erhalten. Mit etwas "Knetmetall" ([dieses Zeug hier](#)) habe ich jeweils eine M10-Edelstahlmutter in die Seestar-Flansche eingematscht, das hält bombenfest. Mit einem Stück Gewinde kann ich jetzt trotz der geringen Gesamthöhe das Teleskop in Waage bringen:



In die abgeschnittenen Stativ-Füsse habe ich - wieder mit der [Metall-Knete](#) - jeweils eine M10-Edelstahlschraube eingearbeitet:



Die Flansche mit der eingebauten M10-Mutter sind montiert:



...und fertig ist das Kistenhöhen-kompatible, nivellierbare Minimalstativ:



## **6) Steuer-Elektronik**

### **6a) Software:**

Das Seestar benötigt als „Steuerungs-Client“ wie gesagt Android oder IOS. Es könnte also mit einem Android-Telefon oder -Tablet gesteuert werden.

Ich habe mich für ein LineageOS auf einem Raspberry PI 4B mit 8 GB RAM entschieden.

Über die GPIO des PI kann dann auch gleich die Steuerung des Linearmotors zum öffnen der Kiste, des Hubmagneten und zum Einschalten des Seestar und des Gehäuselüfters erfolgen.

Bei LineageOS handelt es sich um ein echtes Android, RaspberryPi-OS Programmbibliotheken funktionieren hier **nicht**.

Die Installation auf dem PI ist nicht trivial – alle nötigen Schritte sind in diesem Video erklärt:

<https://www.youtube.com/watch?v=IRFNbLifFu0>

### **Hier die wesentlichen Schritte:**

- 1) auf <https://KonstaKANG.com> das Image für den verwendeten Pi herunterladen  
In meinem Fall (für den Pi4) LineageOS21 (entspricht Android 14)
- 2) Mit dem Programm [Raspberry Pi Imager](#) kann das heruntergeladene Image auf eine SD-Karte installiert werden. Option „Choose OS“ → „use custom“.
- 3) SD-Karte in den Pi stecken und LineageOS booten, Ersteinrichtungsassistenten durchführen
- 4) im Browser des LineageOS die Konstakang-Seite aufrufen. Dort wieder auf die passende Version klicken (LineageOS 21 (Android14) für Pi4
- 5) MindTheGappsxxxxxxxxxxxxxxxxx.zip klicken und bei „Assets“ die zip-Datei herunterladen
- 6) zurück auf die zuletzt aufgerufene KonstaKANG-Seite gehen und xxx-resize.zip herunterladen
- 7) Symbol mit den 9 Punkten (= alle Apps) klicken, dann „Settings“
- 8) in das Suchfeld oben rechts „restart“ eingeben, dann auf „Advanced restart“ klicken.
- 9) Herunter scrollen und „Advanced restart“ einschalten
- 10) die linke obere Desktop-Ecke mit gedrückter linker Maustaste herunterziehen, dann auf das „Power-Symbol“, dann „Restart“ und „Recovery“.
- 11) nach Reboot: „Install“ → „Downloads“ → „lineage-xxx-resize.zip“ → „swipe to confirm“
- 12) „Wipe Dalvik“ → „swipe to wipe“
- 13) „Reboot System“
- 14) kontrollieren, ob das Dateisystem expandiert wurde: Symbol mit den 9 Punkten (= alle Apps) klicken, dann „Settings“. Unter „Storage muß die richtige Größe der SD-Karte angezeigt werden.
- 15) webbrower starten, nach „aptoide“ suchen
- 16) Aptoide-Appstore (apk) von der Aptoide-Seite herunterladen
- 17) linke obere Desktop-Ecke mit gedrückter linker Maustaste herunterziehen, klick auf die apk-Datei --> Installation bestätigen (Settings → Install unknown apps)
- 18) Aptoide-App öffnen → „Search“ → „Device Id vor Android“ herunterladen (es gibt viele davon, die App mit dem großen „I“ und dem kleinen „d“ bei ID funktioniert) → Install (+erlauben)
- 19) die linke obere Desktop-Ecke mit gedrückter linker Maustaste herunterziehen, dann auf das „Power-Symbol“, dann „Restart“ und „Recovery“.
- 20) „Install“ → „Downloads“ → „ MindTheGappsxxxxxxxxxxxxxxxxx.zip“ → „swipe to confirm“
- 21) „Wipe Dalvik“ → „swipe to wipe“
- 22) „Reboot System“
- 23) Unter „Alle Apps“ (= Symbol mit den 9 Punkten) befindet sich jetzt der Google Playstore
- 24) Playstore starten → „Sign in“ → Meldung „This device isn't Play Protect certified“ kommt.
- 25) Link bei „Learn more“ klicken, dort „Device isn't certified“ klicken.

- 26) Herunterscrollen bis „Register your device“ → mit Google-Account anmelden (oder einen erstellen)
- 27) „Updating Google Apps“
- 28) Schritt 24 + 25 wiederholen
- 29) Unter „Alle Apps“ (= Symbol mit den 9 Punkten) „Device Id“ – App starten
- 30) in der Device-ID APP bei „Google Services Framework ID“ auf „Copy“ klicken
- 31) Das Rechteck der 3 Steuersymbole drücken, wieder auf die Google-Anmeldeseite klicken und mit rechter Maustaste & „Paste“ die kopierte Services-Framework-ID in das dafür vorgesehene Feld einfügen, „I' not a robot“ anhaken und „Register“ klicken.
- 32) es dauert eine Weile (10-30 Minuten, evt. Zwischendurch rebooten, dann kann man sich beim Google Playstore anmelden)

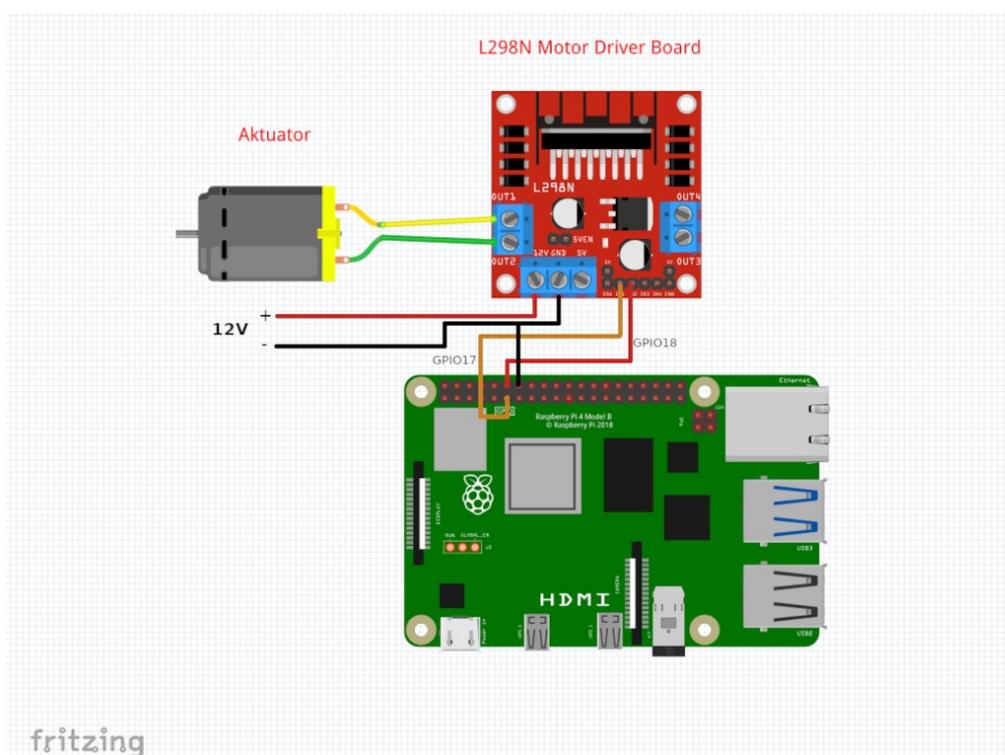
Jetzt können wir uns aus dem Playstore die **Seestar-App** und die App „**SH Script Runner**“ herunterladen.

Die Seestar-APP empfängt auf dem Android-Pi ohne GPS keinen Standort - auch nicht mit Internetzugang. Man kann ja die Location in der Seestar-App händisch auswählen - bei mir gehen die Minuten aber nur bis "39" ?

Die Lösung ist eine Mock-GPS-APP wie z.B. "**Fake-GPS**". Einfach aus dem Playstore laden, in den Entwickleroptionen unter "Standort --> App für simulierte Standorte auswählen" die App eintragen und den gewünschten Standort auf der Karte der App wählen.

Die Ansteuerung der GPIO ist mit Android über die üblichen Raspberry-Bibliotheken **NICHT** möglich. Eine Lösung ist die Verwendung von [sysfs](#). Hierzu werden in der App „SH Script Runner“ folgende Scripte angelegt (beim Abspeichern jeweils „use Root“ anhaken!)

Zur Ansteuerung des Linear-Motors, der den Deckel öffnet habe ich den [L298 Motortreiber](#) verwendet



## Die Scripte dazu lauten:

### Observatorium öffnen (L298N IN1=HIGH IN2=LOW)

```
su
echo 17 > /sys/class/gpio/export
echo 18 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio17/direction
echo out > /sys/class/gpio/gpio18/direction
echo 1 > /sys/class/gpio/gpio17/value
echo 0 > /sys/class/gpio/gpio18/value
```

### Observatorium schliessen (L298N IN1=LOW IN2=HIGH)

```
su
echo 17 > /sys/class/gpio/export
echo 18 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio17/direction
echo out > /sys/class/gpio/gpio18/direction
echo 0 > /sys/class/gpio/gpio17/value
echo 1 > /sys/class/gpio/gpio18/value
```

### STOP (L298N IN1=LOW IN2=LOW)

```
su
echo 17 > /sys/class/gpio/export
echo 18 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio17/direction
echo out > /sys/class/gpio/gpio18/direction
echo 0 > /sys/class/gpio/gpio17/value
echo 0 > /sys/class/gpio/gpio18/value
```

Der Hubmagnet-Einschalter wird über eine Relaisplatine betätigt.

Der Magnet muss anziehen und nach einer Zeit wieder abfallen.

Mit sysfs ist der Code dafür ein bisschen "weird", denn die GPIO muss erst "unexported" werden um einen neuen Zustand anzunehmen:

### Seestar einschalten

```
su
echo 22 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio22/direction
echo 1 > /sys/class/gpio/gpio22/value
echo 22 > /sys/class/gpio/unexport
sleep 5
echo 22 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio22/direction
echo 0 > /sys/class/gpio/gpio22/value
```

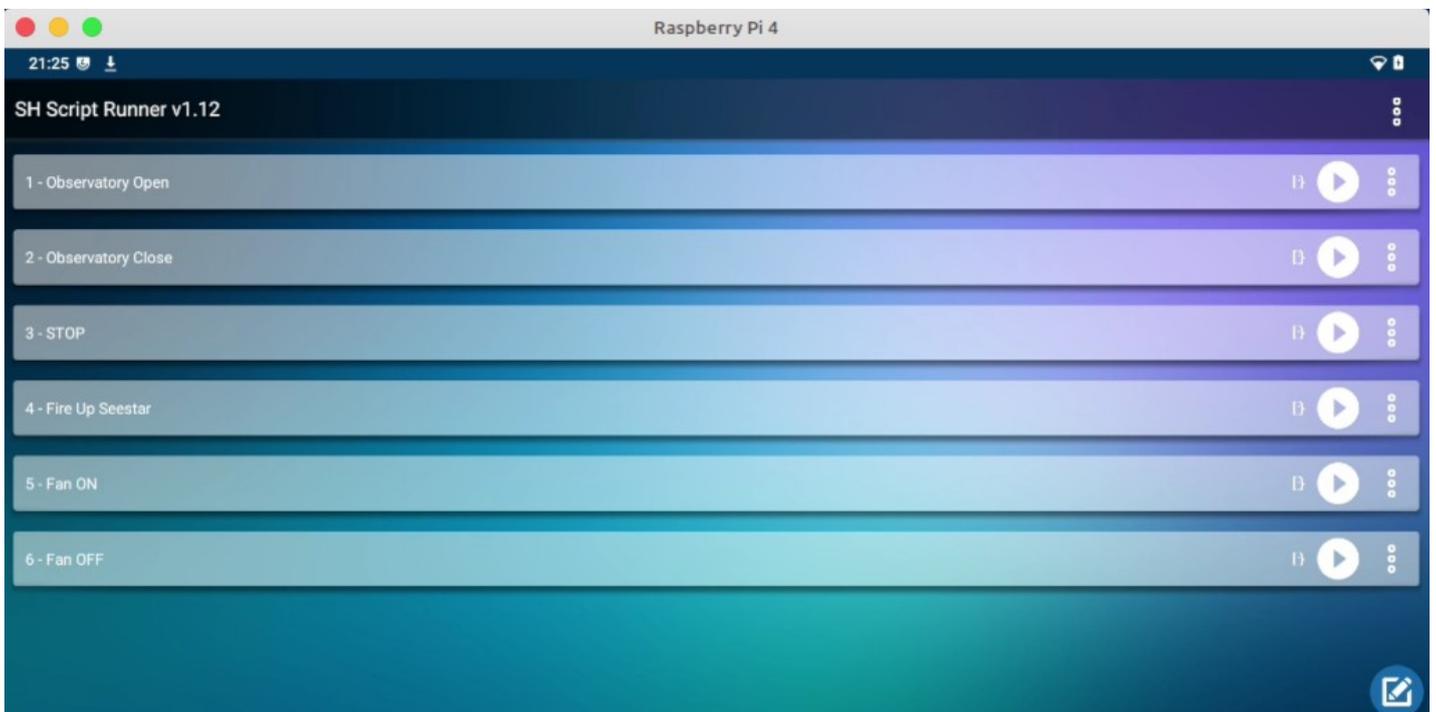
## Gehäuselüfter ein:

```
su  
echo 24 > /sys/class/gpio/export  
echo out > /sys/class/gpio/gpio24/direction  
echo 1 > /sys/class/gpio/gpio24/value
```

## Gehäuselüfter aus:

```
su  
echo 24 > /sys/class/gpio/export  
echo out > /sys/class/gpio/gpio24/direction  
echo 0 > /sys/class/gpio/gpio24/value
```

So sieht die dann komplette GUI aus:

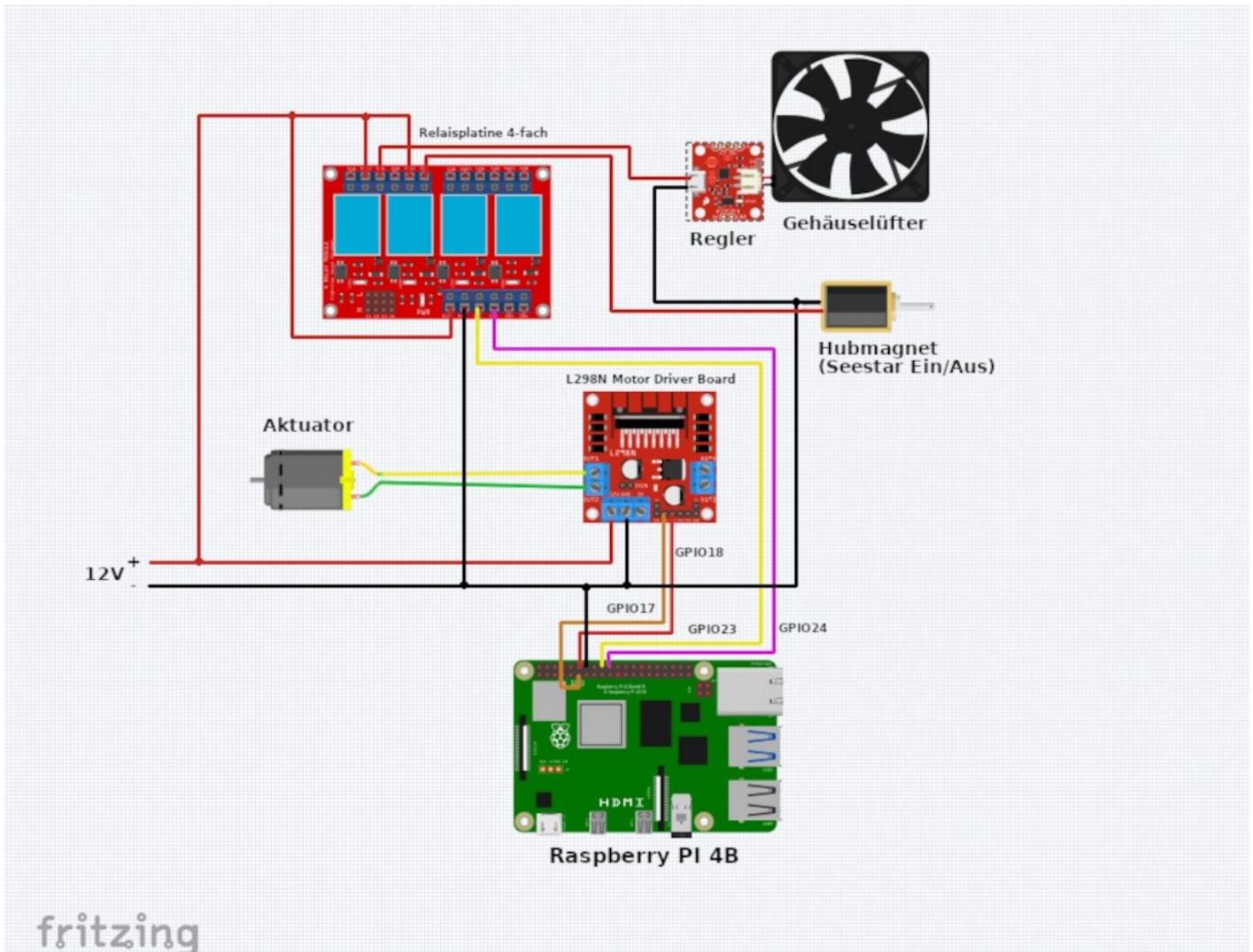


## 6b) Steuerungs-Hardware und die Verkabelung der Komponenten:

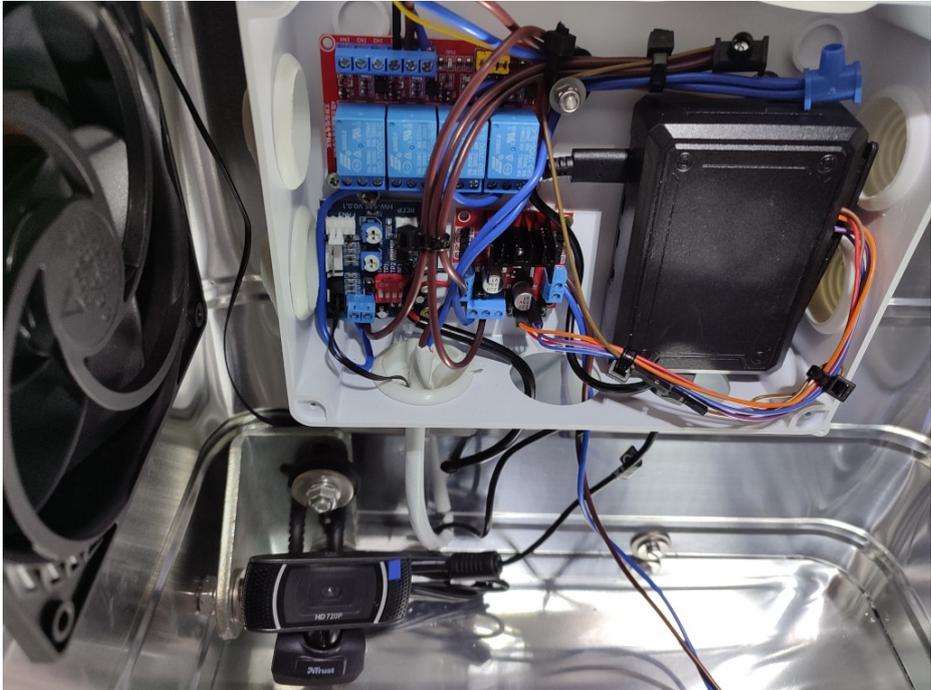
Der Linearmotor läuft über ein LN298N Motortreiber-Modul

Der Gehäuselüfter läuft über ein Reglermodul

Hubmagnet-Einschalter und Lüfter werden über eine Relaisplatine geschaltet



Die verkabelte Elektronik sitzt in einem geschützten Gehäuse:



Das Seestar braucht WLAN, der Pi käme auch mit LAN klar.

Der Einfachheit halber habe ich mich für einen Accesspoint im Inneren der Box entschieden, so stört die Alukiste nicht und der Empfang ist immer top.

Wer so eine Kiste im Aussen-Einsatz auf dem Acker betreiben möchte, kann natürlich auch einen LTE-Router verwenden.

Ich habe mich für eine Fritzbox7530AX entschieden, denn diese lässt sich nahtlos als MESH-Komponente an die vorhandene 7590 anbinden. Der Accesspoint hat dann automatisch die gleiche SSID, das Seestar und der PI sind im selben Netzwerk und im Station-Modus läuft dann auch der Seestar-Stream im LAN.

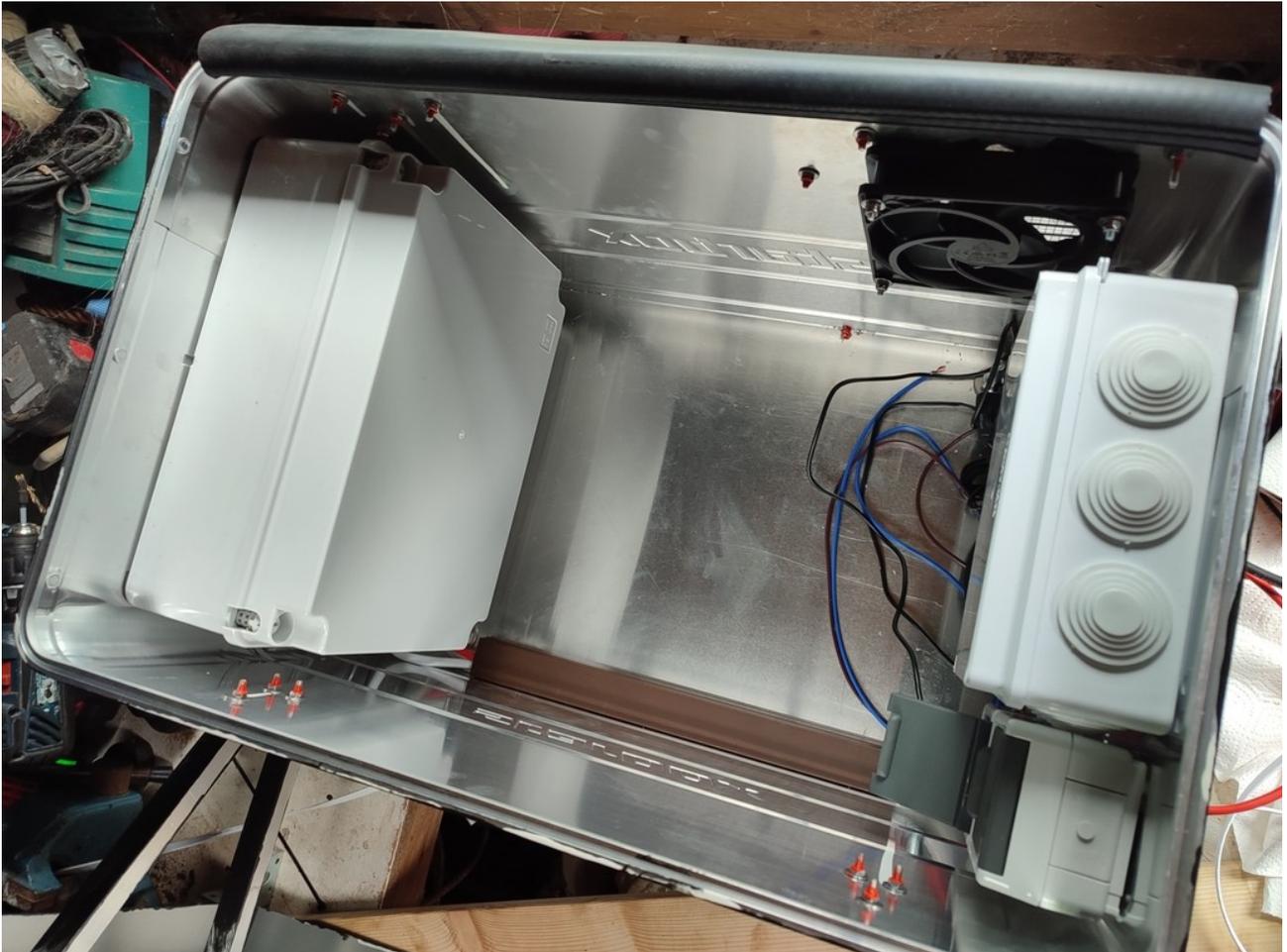
Die Verbindung der beiden Boxen ist nahezu konfigurationsfrei, man belässt die 7530AX einfach bei den Werkseinstellungen und drückt erst an der 7590 und dann an der 7530 die WPS-Taste. Damit ist die 7530AX bereits als MESH-Accesspoint konfiguriert.



Im Fritzbox-Gehäuse habe ich zwei Steckdosen untergebracht: eine für die Fritzbox und zusätzlich eine für das Netzteil für alle 12V-Geschichten (Aktuator/Motortreiber, Lüfter, Relaisplatine, Hubmagnet-Einschalter) und das 12V-Netzteil hatte auch noch Platz.

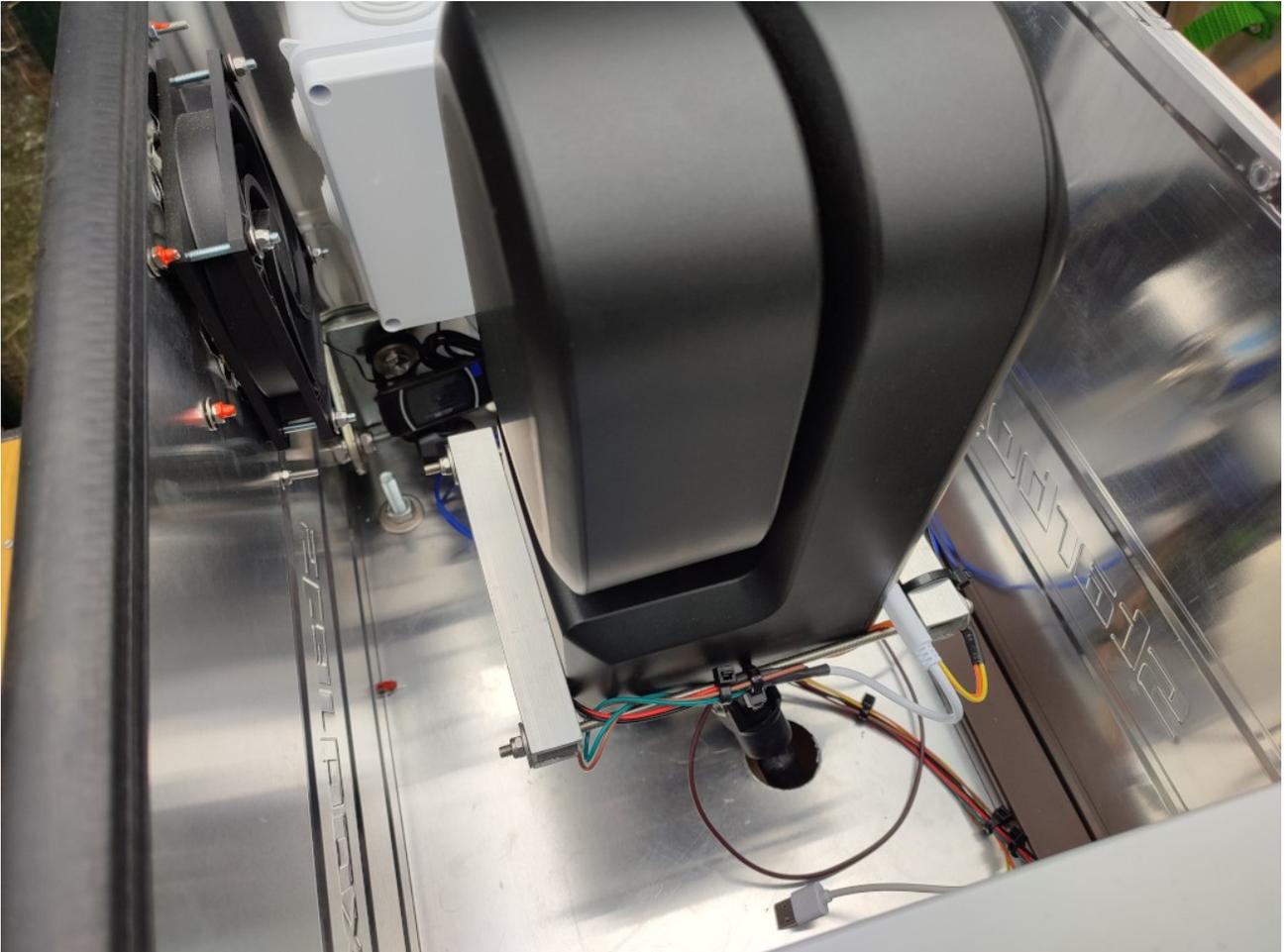
Auf der anderen Seite habe ich auch noch zwei Steckdosen montiert (VDE beachten !!!)

Mit Deckel drauf sieht es schon viel aufgeräumter aus:



Der Kistenboden ist recht dünn und da er nicht aufliegt, federt er etwas.

Ich habe mich deshalb dazu entschlossen, das Seestar auf seine eigenen Beine zu stellen und habe mit einem Kreisbohrer 3 Löcher in den Kistenboden gebohrt:



Damit das Seestar beim Transport der Kiste trotzdem an Ort und Stelle bleibt, habe ich an jedes Bein eine [1/2"-Schraubrohrschele](#) befestigt.

Die Schellen sind mit dem Kistenboden mit jeweils einem [Stangen-Endlager](#) verbunden. So ist das Seestar gut befestigt und bleibt trotzdem höhenverstellbar:



Das Seestar benötigt zur Legitimation der Netzwerkverbindung eine Bluetooth-Verbindung.

Dieser USB-Dongle funktioniert mit LineageOS „out of the Box“:

<https://www.amazon.de/dp/B08MWRC4BG>

## **7) Fernzugriff:**

Alle mir bekannten Remote-Clients (außer Teamviewer, der in der kostenlosen Version die Sitzungsdauer einschränkt), laufen entweder nicht auf Android-Systemen (NoMachine gibt es nur als „Client“) oder taugen nicht für den „unbeaufsichtigten Zugriff“ (z.B. Anydesk).

Ich habe mich deshalb für einen Zugriff über [scrcpy](#) entschieden. Das geht so:

### Auf dem Android-Pi:

- 1) Im Android die Entwickleroptionen aktivieren (7x die build-Nummer unter "System--> über das Tablet) klicken
- 2) In den Entwickleroptionen "Debugging über WLAN" aktivieren
- 3) Unter System --> Raspberry Pi settings "ADB" aktivieren

### Auf dem Computer:

- 1) [scrcpy](#) herunterladen
- 2) [sdk-platform-tools](#) herunterladen
- 3) beide Ordner entpacken und den Inhalt des scrcpy-Ordners in den platform-tools order kopieren
- 4) über ein Terminal folgendes nacheinander ausführen:

```
adb connect IP-des-PI:5555
```

```
scrcpy --video-encoder=OMX.google.h264.encoder -m 1024 --max-fps 16
```

Ich habe mir beides in ein Script gepackt, so kann ich das Observatorium mit einem Klick erreichen.

Das ganze kann per VPN und NoMachine, RDP, o.ä. über den Host-PC auch mobil vom Handy oder von anderen Rechnern an anderen Standorten weltweit bedient werden.

Wenn alles glatt gelaufen ist, sollte das Mini-Observatorium jetzt funktionieren.

Viel Spaß!

Video Bedienung: [https://bramborka.com/fp2/OBS1\\_1.mp4](https://bramborka.com/fp2/OBS1_1.mp4)

Video Deckel schliessen: [https://bramborka.com/fp2/OBS2\\_1.mp4](https://bramborka.com/fp2/OBS2_1.mp4)

