

## Weltraumabenteuer beginnen auf der Erde: Astro-Pi - ein Raspi mit SenseHat - hilft dabei!

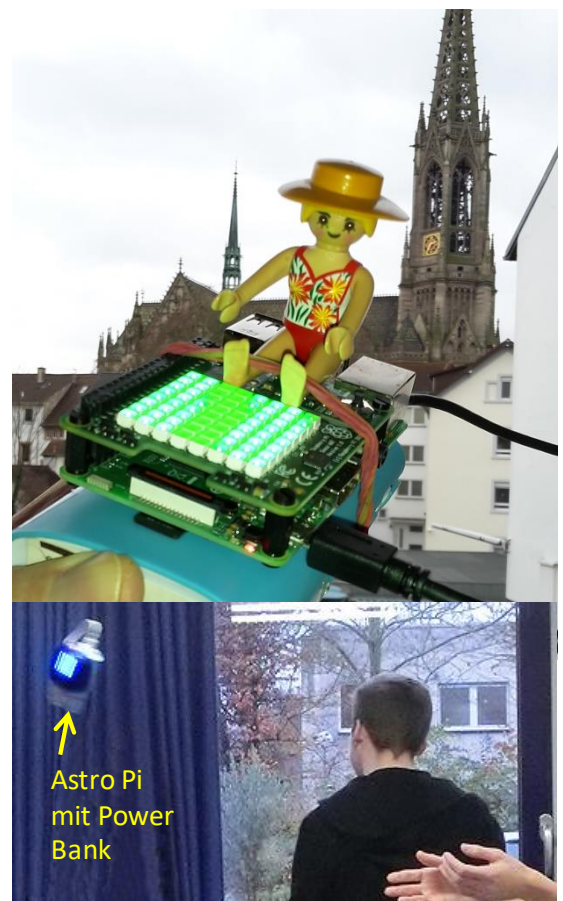
Reisen in den Weltraum sind in der Phantasie immer möglich. Wenn es um die Verwirklichung geht, ist es gut, bald mit dem Training zu beginnen. Dazu gehört es, Vorstellungen von dem zu entwickeln, was uns erwartet, wenn wir unseren Planeten verlassen.

Eine Möglichkeit sind Simulationen. Einigermaßen komfortabel lassen sich digitalen Experimente mit dem sogenannten „Astro Pi“ verwirklichen. Das ist ein Einplatinencomputer mit einem Sensor- und Anzeigeaufsatz, „SenseHat“ genannt. Der Astro Pi hat zusammen mit einem passenden Akku (Power Bank) Platz auf einer Handfläche. (Weitere Infos zum Astro-Pi: [Siehe Link im Bild oben!](https://www.rasberrypi.org/learning/astro-pi-guide)) Mit seiner Hilfe lassen sich die Verhältnisse beim Parabelflug veranschaulichen: Während des Fluges – allein unter Wirkung der Schwerkraft – herrscht in und auf einem entsprechenden Flugobjekt Schwerelosigkeit, weil es der Schwerkraft ungehindert folgt. Wichtig: Das Fehlen von zusätzlicher Antriebs- und Bremskraft!

Die Gewichtskraft kann also durch den freien Fall aufgehoben werden. Dieser freie Fall setzt tatsächlich schon ein, sobald der zusätzliche Antrieb fehlt, also auch, wenn das Objekt noch gar nicht fällt, sondern nur von der Schwerkraft gebremst hochsteigt. Beim Hin- und Herwerfen mit einem Astro-Pi zeigen die LEDs das deutlich an.

Die Programme, die das Zusammenspiel von Raspberry und SenseHat steuern, sind in Python geschrieben und relativ einfach, siehe Bild unten.

(Eine Python-Entwicklungsumgebung für den Raspberry Pi erhält man zusammen mit dem kostenlosen Betriebssystem, wenn man dieses herunterlädt.)



Die in „Parabelflug.py“ gewählte Darstellung von Schwerelosigkeit, normaler Schwerkraft oder erhöhter Krafteinwirkung aufgrund zusätzlicher Beschleunigungs- oder Abbremsvorgänge nutzt das LED-Display auf dem SenseHat.

```
Parabelflug.py - \\ASTROPi\\pi\\Parabelflug.py (3.6.3)
File Edit Format Run Options Window Help
from sense_hat import SenseHat
import math
sense = SenseHat()
while True:
    a3d = sense.get_accelerometer_raw()
    ax=a3d['x'];ay=a3d['y'];az=a3d['z']
    aGesamtBetrag = math.sqrt(float(ax)*float(ax)+float(ay)*float(ay)+float(az)*float(az))
    if aGesamtBetrag >=1.05:r=255;g=0;b=0
    if aGesamtBetrag <= 0.95:r=0;g=0;b=255
    if aGesamtBetrag > 0.95 and aGesamtBetrag < 1.05:r=0;g=255;b=0
    print(aGesamtBetrag)
    sense.clear([r, g, b])
```

Wir haben folgende Farbcodierung gewählt:

- Schwerelosigkeit (bzw. deutlich verminderte Schwerkraft): Blau,
- Schwerkraftwerte ungefähr wie auf der Erde: Grün,
- Starke Brems- und Beschleunigungskräfte: Rot.

In gewisser Weise als Gegenteil lässt sich verdeutlichen, wie man im Weltraum künstliche Schwerkraft erzeugen kann. Das ist wichtig, denn für Astronautinnen und Astronauten ist es ungesund, zu lange in der Schwerelosigkeit zu leben, unter anderem, weil dann wichtige Muskeln verkümmern. Eine Lösungsmöglichkeit: Die Raumstation wird wie ein großes Karussell in Drehung versetzt. Die Fliehkraft übernimmt dann die Rolle der Schwerkraft, welche dann natürlich nach außen gerichtet ist. Das kann man mit einem Astro Pi zeigen, der auf ein Fahrrad-Rad montiert wird. Dieses wird dann waagrecht eingespannt, so dass von der Achse weg nach außen keine Kräfte wirken, solange es sich nicht dreht. Wenn es dann mit der richtigen Drehzahl gedreht wird, dann sieht man an den LEDs ähnlich wie beim Parabelflug, wann das richtige Maß an „Schwerkraft“ erzeugt wird.



An der IGS LuGa haben wir die radialen (nach außen gerichteten) Beschleunigungswerte des Raspis mit Grafiken auf dem Display eines Laptops in der „Bodenstation“ zusätzlich anschaulich zu machen versucht. So können die Schülerinnen und Schüler bildlich unterstützt Erfahrungen damit machen, dass man für die passende künstliche Schwerkraft eine bestimmte Drehzahl braucht.

Der Raspberry Pi muss sich sowohl auf einer Parabelflugbahn oder auf einem Karussell ohne Kabel bewegen können, darum wird er jeweils mit einem Powerpack (Akku) ausgestattet. Für die Bodenstation des Schwerkraftkarussells zur Anzeige der Animation ist zusätzlich eine Netzwerkanbindung notwendig. Und die muss kabellos sein. Wir machen das über einen kleinen WLAN-Router. Dazu mehr auf den folgenden Seiten.

Hier noch einige Anmerkungen zum Raspberry Pi und der Software, welche für die Vorarbeiten zu diesem Skript verwendet wurde.

(Alle Handgriffe können hier nicht erklärt werden. Gute Hilfestellungen findet man in Foren im Netz, ggf. auch unter Kolleginnen und Kollegen.)

Es handelt sich um einen Raspberry Pi 3 B.

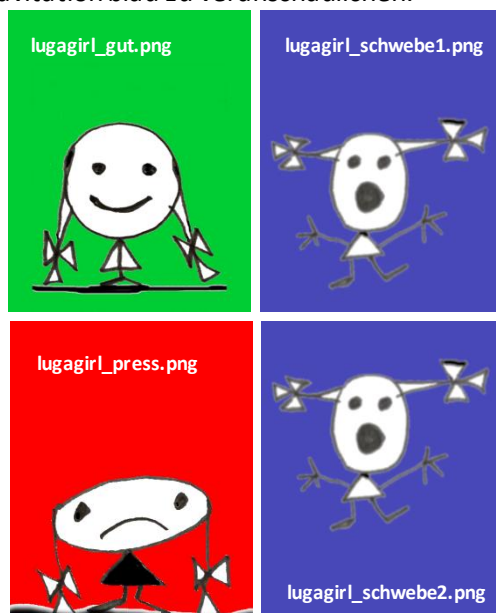
(Es gibt auch Vorgänger mit entsprechend anderen Bezeichnungen.)

Als Betriebssystem wurde die folgende Version als Image-Datei auf eine 8 GB große Micro-SD-Karte (welche als Festplatte dient) geschrieben:

Raspbian Stretch with desktop  
Image with desktop based on Debian Stretch  
Version: November 2017  
Release date: 2017-11-29

Diese Image-Datei wurde heruntergeladen von:  
<https://www.raspberrypi.org/downloads/raspbian/>.  
Das kostenlose Programm „Image Writer“ diente zum Übertragen auf die SD-Karte.

Die Bilder unten werden im Programm „girl-g-rot.py“ verwendet, um hohe künstliche Gravitation rot, normale grün sowie geringe künstliche Gravitation blau zu veranschaulichen.



```
KuenstlicheSchwerkraft.py - \\ASTROPI1\pi\KuenstlicheSchwerk...
File Edit Format Run Options Window Help

from tkinter import *;from sense_hat import SenseHat
import time, math
global sense
sense = SenseHat()
def mess():
    a3d = sense.get_accelerometer_raw();az=a3d['z'];print(az)
    if float(az)>= 1.1:
        r=255;g=0;b=0;sense.clear([r, g, b])
        tkFenster.after(10,press)
    if float(az)>= 0.8 and float(az) <= 1.1:
        r=0;g=255;b=0;sense.clear([r, g, b])
        tkFenster.after(10,gut)
    if float(az)<= 0.8:
        r=0;g=0;b=255;sense.clear([r, g, b])
        tkFenster.after(10,schwebel)
def schwebel():
    labelBild.config(image=bild_schwebel)
    time.sleep(0.2)
    tkFenster.after(10,schwebe2)
def schwebe2():
    labelBild.config(image=bild_schwebe2)
    time.sleep(0.2)
    tkFenster.after(10,mess)
def gut():
    labelBild.config(image=bild_gut)
    time.sleep(0.2)
    tkFenster.after(10,mess)
def press():
    labelBild.config(image=bild_press)
    time.sleep(0.2)
    tkFenster.after(10,mess)
tkFenster = Tk()
tkFenster.geometry('600x600')
bild_schwebel=PhotoImage(file="lugagirl_schwebel.png")
bild_schwebe2=PhotoImage(file="lugagirl_schwebe2.png")
bild_press=PhotoImage(file="lugagirl_press.png")
bild_gut=PhotoImage(file="lugagirl_gut.png")
frameBild = Frame(master=tkFenster)
frameBild.place(x=0, y=0, width=600, height=600)
labelBild = Label(master=frameBild,image=bild_gut)
labelBild.pack()
tkFenster.after(1000,schwebel)
tkFenster.mainloop()

Ln: 44 Col: 0
```

# Den Astro Pi von der Leine lassen

Der Raspberry Pi muss sich auf einem Karussell ohne Kabel zum Bildschirm bzw. Laptop bewegen. Ob man entsprechende Programme mit differenzierter Unterstützung im Unterricht (mit-)entwickeln lässt oder sie als fertiges Unterrichtsmaterial vorgibt, hängt wesentlich natürlich von der Lerngruppe ab.

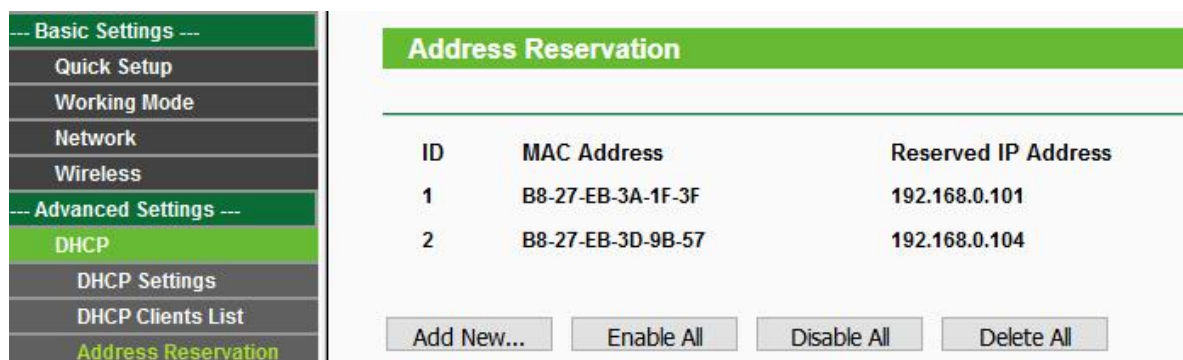
## Einbindung in ein WLAN

Eine andere Schwierigkeit muss in der Regel von der Lehrerin oder dem Lehrer im Vorfeld ausgeräumt werden: Damit die Sensordaten und die Befehle für die Animationen auf dem Raspi aktuell ausgelesen werden können, muss er drahtlos zugänglich gemacht werden. Dazu kann er über WLAN mit einem Computer oder Smartphone etc. verbunden werden. Das macht in der Regel Probleme, es sei denn man ist zu Hause, wo man den Raspberry Pi zunächst mit Hilfe von Tastatur, Maus und Bildschirm (mit HDMI-Anschluss!) wie jeden anderen Computer einbinden kann. In der Schule und im Freiland ist das in der Regel nicht der Fall. Dort ist entweder nur ein geschütztes oder gar kein WLAN vorhanden.

Wir haben das Problem folgendermaßen gelöst: Ein kleiner WLAN-Router (TP-Link TL-WR702N, ca. 20€) wird zwecks Stromversorgung an einen Laptop über Micro-USB-Kabel angeschlossen. Er stellt dann ein WLAN zur Verfügung, in das sowohl der Laptop als auch der Raspberry eingebunden werden.

Damit man die Anmeldung ins WLAN nicht jedes Mal neu machen muss und man feste IP-Adressen hat, zu deren Bedeutung wir gleich kommen, haben wir im Menü des Routers den einzelnen Rasperrys feste IP-Adressen zugewiesen. Dazu sind mehrere Schritte notwendig, die man sich möglichst einmal konkret zeigen lässt, wenn man darin keine Übung hat, was in der Regel der Fall sein sollte:

- Erst meldet man sowohl den Raspberry über sein WLAN-Menü (oben rechts auf seinem Desktop) normal an als auch den Laptop (unten rechts auf dem Desktop). Die Zugangsdaten (WLAN-Name und -Schlüssel) stehen auf dem Router-Gehäuse. So erhält der Raspi vom Router eine (auf Dauer aber nicht feste) IP-Adresse mittels „dhcp“.
- Wenn beide Geräte im WLAN sind, lässt man auf dem Laptop einen IP-Scanner laufen, z.B. den „Advanced IP Scanner“, der kostenlos im Netz zur Verfügung gestellt wird. Er ermittelt die MAC-Adresse des Raspis. Damit das nicht zu lange dauert, schaut man besser im Menü des Routers nach, welcher Adressbereich für „dhcp“ freigegeben ist.
- Mit der MAC-Adresse, die im WLAN-Modul des Raspis werkseitig fest eingestellt ist, lässt sich dann im Router eine - in Zukunft feste - IP-Adresse wählen, siehe Bild:



Erfolgreich kann auch die Nutzung von einem Hotspot sein, den man durch ein Smartphone zu Verfügung stellt. Hier muss man im Einzelfall testen, wie weit die Verbindung reicht. Und jemand muss sein Smartphone dafür zu Verfügung stellen. Unseres Wissens nach lassen sich dann leider keine festen IP-Adressen einstellen.

## Ferngesteuerten Raspi-Desktop mit VNC-Server und VNC-Viewer auf den Notebook-Bildschirm bringen

Wenn die WLAN-Einbindung steht, dann muss auf dem Raspi ein VNC-Server zum Laufen gebracht werden \* und auf dem Laptop ein VNC-Viewer. Um den VNC-Viewer mit dem Raspi zu verbinden, muss man dessen IP-Adresse für die Anmeldung am Server kennen. \*\*

Der Server schickt dann die Desktop-Ansicht des Raspis an den Viewer, welcher diese auf dem Laptop (wie in Bild rechts) sichtbar macht und auch seine Bedienung erlaubt. So lassen sich Programme auf dem Raspi starten und die Messwerte, Grafiken etc. anzeigen. Das gilt auch für Fotos (falls der Raspi eine Kamera hat), aber leider nicht für Filme oder Video-Liveansichten.

Das Bild rechts zeigt die Ansicht eines Raspi-Desktops, der mit einem VNC-Viewer auf einen Notebook-Bildschirm geholt worden ist. Mit der Tastatur und der Maus des Notebooks wurde auf dem Bild gerade ferngesteuert ein Ordner geöffnet und die dort gespeicherten Programme könnten gestartet werden.



\* Das geht einfach über ein Menü, das über den Desktop des Raspis zugänglich ist und bei den „Schnittstellen“ lediglich ein Anklicken erfordert.

\*\* Ins LX-Terminal den Befehl „ifconfig“ eingeben oder eine IP-Scanner nutzen – siehe Seite 4.



# Wege und Umwege, einen Raspberry Pi 3 mit Stretch-Betriebssystem zu steuern und einzustellen

## 1. Raspberry als normaler Rechner und das Zusammenspiel „Laptop – Raspi“

Der einfachste Weg, einen Raspi zu nutzen und Einstellungen vorzunehmen, ist, ihn als einen eigenständigen Rechner zu nehmen und ihn mit einem eigenen HDMI-Bildschirm, einer eigenen Tastatur und einer eigenen Maus auszustatten. Dann ist der Umgang weitgehend unkompliziert, wenn man es geschafft hat, eine Image-Datei auf die SD-Karte zu kopieren. Die Stromversorgung gelingt über ein Micro-USB-Kabel an einem entsprechenden Ladegerät oder einem Akku (Power Bank).

Bei dieser Art von Zugang kann der WLAN-Zugang wie oben beschrieben eingestellt werden und das Zusammenspiel von Raspberry und Laptop mittels VNC-Server auf dem Raspi und VNC-Viewer auf dem Laptop eingestellt und geübt werden.

Im Folgenden sind noch einige Informationen zusammengetragen, die nicht in aller Ausführlichkeit dargelegt werden können, zumindest aber wesentliche Stichworte zum Recherchieren liefern.

## 2. Zugriff über Smartphone oder Tablet

Hatte man zuvor Gelegenheit, auf den Raspi zuzugreifen, konnte man mit Hilfe des Befehls „sudo nano /etc/dhcpd.conf“ in der Datei „dhcpd.conf“ folgende Zeilen am Ende einfügen:

```
interface usb0
allow-hotplug usb0
iface usb0 inet static
static ip_address=192.168.42.42
```

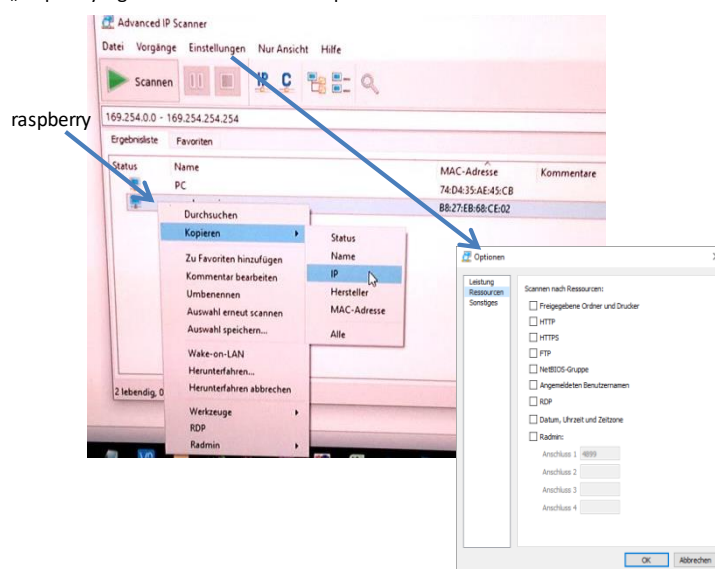
Später kann man dann mit einem Smartphone, auf welchem ein VNC-Viewer installiert ist, über ein USB-Kabel auf den Desktop des Raspis zugreifen. Wichtig ist dabei, dass in den Einstellungen des Smartphones „USB-Tethering“ gewählt wird, was allerdings nur möglich ist, wenn der Raspi schon hochgefahren und über USB-Kabel am Smartphone angeschlossen ist. Konnte man nicht auf den Raspi zugreifen, lässt sich der Weg über Smartphone, USB-Tethering und IP-Scanner gehen, was relativ kompliziert ist, den Aufwand aber lohnt, wenn man öfter mit Raspberry-Pis arbeitet, siehe Abschnitt 4 auf der nächsten Seite.

## 3. SSH über ein LAN-Kabel

Wenn der HDMI-Bildschirm fehlt oder aus anderen Gründen kein erster Kontakt mit dem Raspi aufgenommen werden kann, gibt es folgende Möglichkeiten über ein LAN/Ethernet-Kabel und die SSH-Schnittstelle Verbindung aufzunehmen mittels eines Terminal-Programms wie „Putty“. Bei der neuen Betriebssystemversion „Stretch“ (s.o.) ist der SSH-Zugang allerdings aus Sicherheitsgründen gesperrt und kann nur über einen Trick geöffnet werden. Hier das Vorgehen in Kurzform:

**Raspi 3 mit „2017-11-29-raspian-stretch.img“ über „ssh“ für „vnc“ zugänglich machen**

1. Img auf SD schreiben, Warnungen und Fehlermeldungen ggf. ignorieren.
2. Auf der fertigen SD-Karte in „boot“ eine leere Datei „ssh.txt“ erstellen, dann in deren Namen „.txt“ löschen und trotz Warnung die Umbenennung durchführen. (Achtung: Bei jedem Hochfahren des Raspis wird die Datei „ssh“ gelöscht und muss ggf. neu erstellt werden.)
3. SD-Karte in den Raspi 3 einsetzen, Ethernetkabel anschließen, Raspi an Strom anschließen.
4. Mit „Advanced\_IP\_Scanner“ die IP-Adressen scannen im Bereich 169.254.0.0 bis 169.254.254.254. Das kann dauern. Über „Einstellungen => Optionen“ bei „Scannen nach Ressourcen“ alle Häkchen herausnehmen, dann geht es zügiger. Wenn „raspberrry“ angezeigt wird, mit rechter Maustaste auf „raspberrry“ gehen und IP-Adresse kopieren.

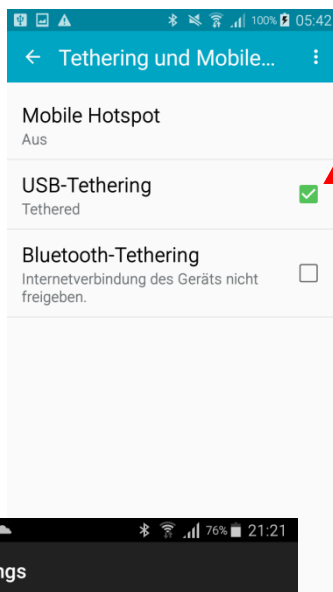


5. Putty-Terminalprogramm starten, IP-Adresse einfügen, trotz Sicherheitswarnung verbinden, als user „pi“ und mit password: „raspberrry“ anmelden (ohne Anführungszeichen).
6. Über den Kommandozeilen-Befehl „sudo raspi-config“ zu den „Interfacing Options“ gelangen, „enable SSH“ und „enable VNC“ klicken, dann zu „OK“ und „Finish“ mit Pfeil- und Tab-Tasten navigieren und mit „Enter“ abschließen.
7. „sudo reboot“ ins Terminal eingeben, mit „Enter“ den Raspi neu starten lassen.
8. Eine VNC-Verbindung auf dem Laptop mit der oben gefundenen IP-Adresse konfigurieren,
9. ggf. VNC-Modus „nur lesen“ herausnehmen,
10. Verbindung herstellen, dann den user „pi“ mit dem Kennwort „raspberrry“ (jeweils ohne Anführungsstriche) anmelden.

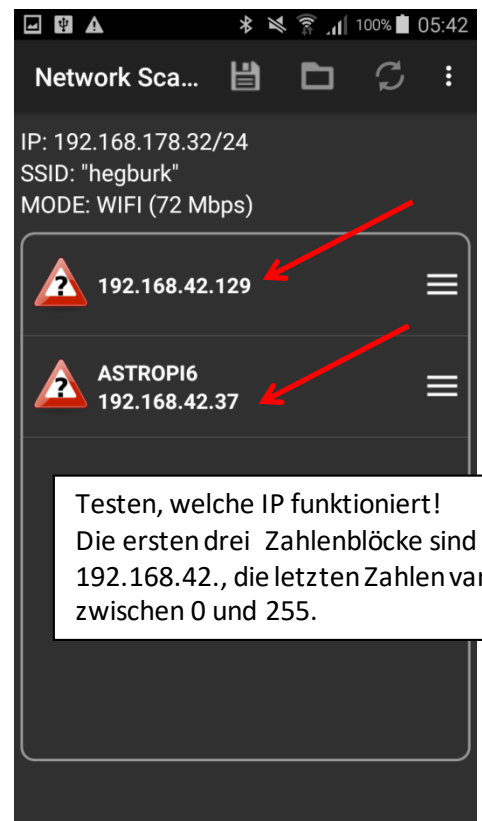
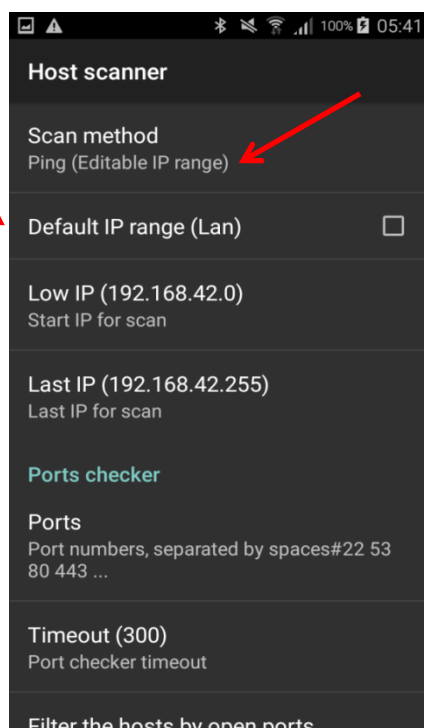
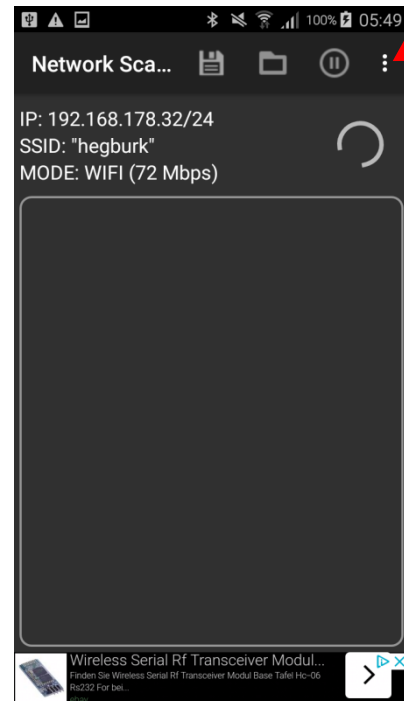
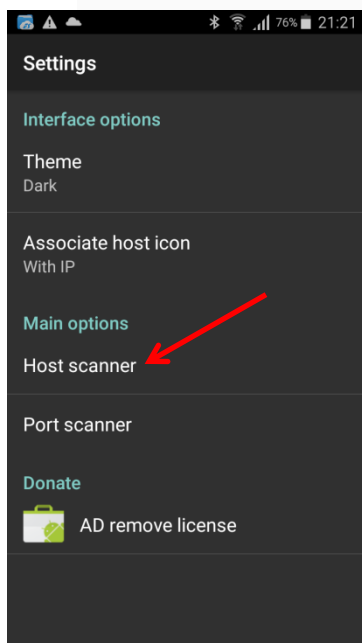
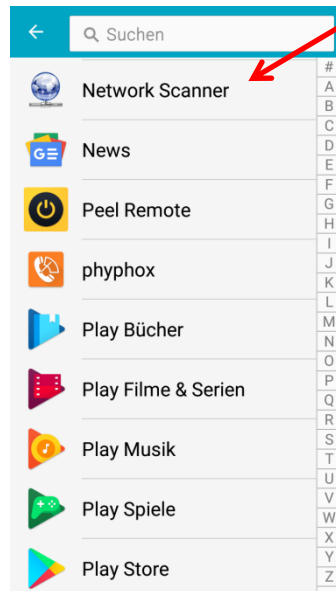
#### 4. Ermitteln der IP-Adresse für VNC mittels Network Scanner

Wichtig: Auf richtiges USB-Kabel achten! Es gibt z.T. „nur Ladekabel“, die gehen für das USB Tethering nicht.

So lässt sich über das Smartphone mit dem Network Scanner aus dem Playstore die IP-Adresse für das USB-Tethering ermitteln, das über VNC-Viewer Zugang zum Raspi ermöglicht:



Geht erst, wenn das USB-Kabel zwischen großem USB-Anschluss am Raspi und Micro-USB-Buchse am Smartphone eingesteckt ist und muss jedes Mal neu gesetzt werden, wenn das Kabel entfernt wurde.



Testen, welche IP funktioniert!  
Die ersten drei Zahlenblöcke sind immer 192.168.42., die letzten Zahlen variieren zwischen 0 und 255.