

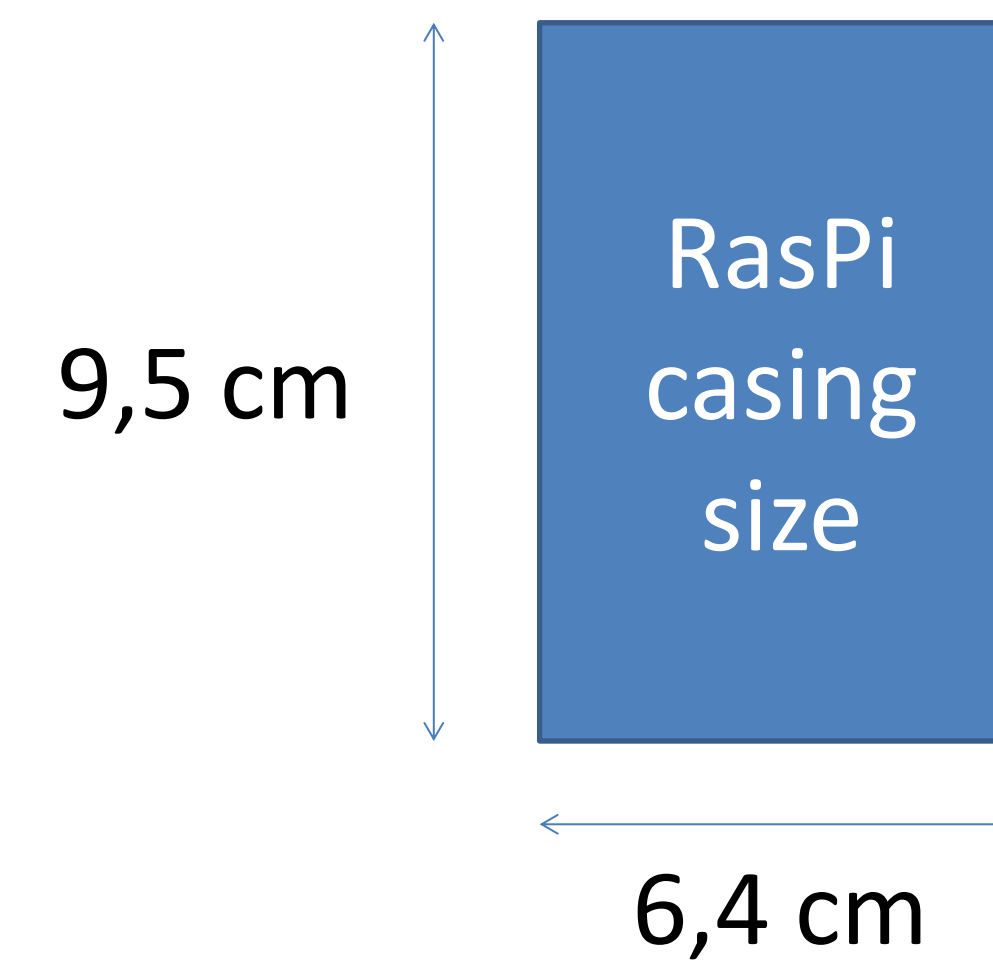
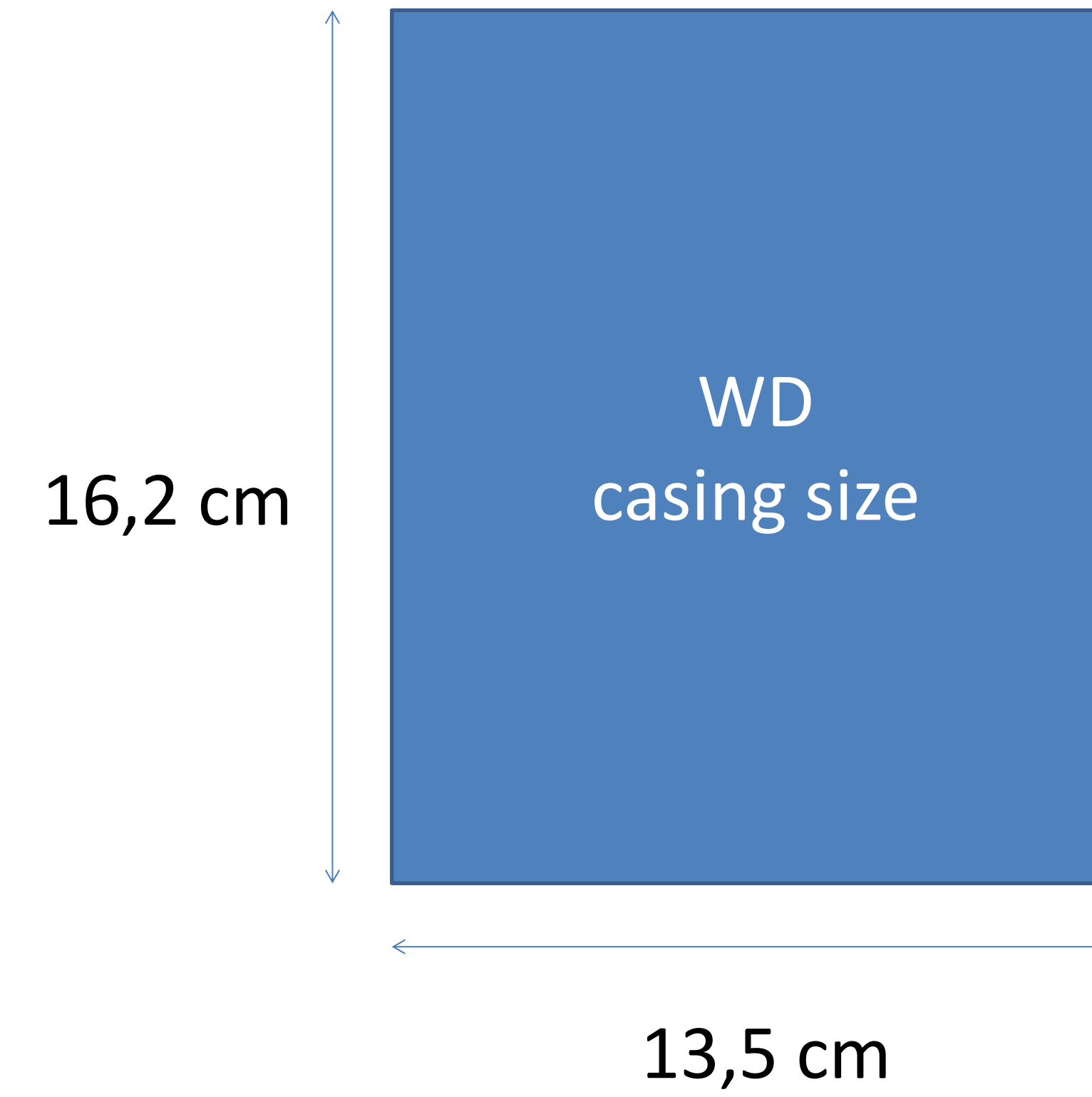
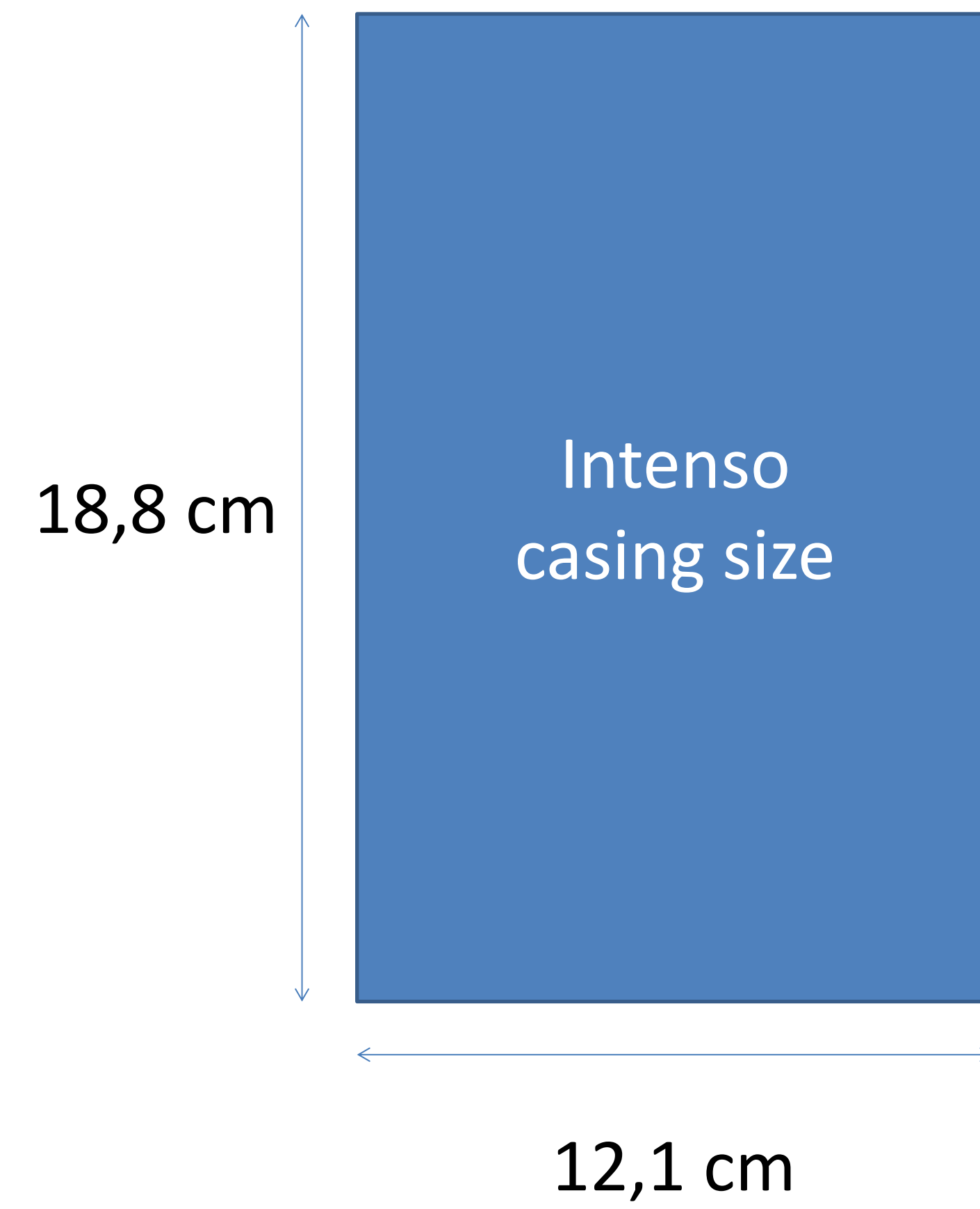
von

Timo Benzel

Using a Raspberry Pi 2 V1.1
and two 4 TB HDDs
to create a NAS drive and a printer server



Mögliche Anordnung
mit Pi
Intenso 4TB und
WD 4TB



Mögliche Anordnung
mit Pi
Intenso 4TB und
WD 4TB

HW Aufbau

- Kombination von WD 4TB-HDD und Intenso 4TB-HDD ist eher unglücklich
- Anderes Inventar: WD 3,5“ 4TB-HDD (Desktop-Gehäuse)
→ Bestelle eher nochmals eine dieser Art und packe die zwei in ein externes Festplatten Gehäuse
- **Achtung:** Controller der externen Festplatte muss 4 TB verwalten können (viele können nur 2 TB!)

HW Aufbau

- Pi und Festplatten sitzen auf Winkel



- Lockblechsteifen (s.u.) halten Pi und Festplatten vom überkippen zurück
- Lockblechsteifen und Winkel sind mit Schrauben an Brett befestigt
- Überziehe Lochblechstreifen mit Schrumpfschlauch (rot)



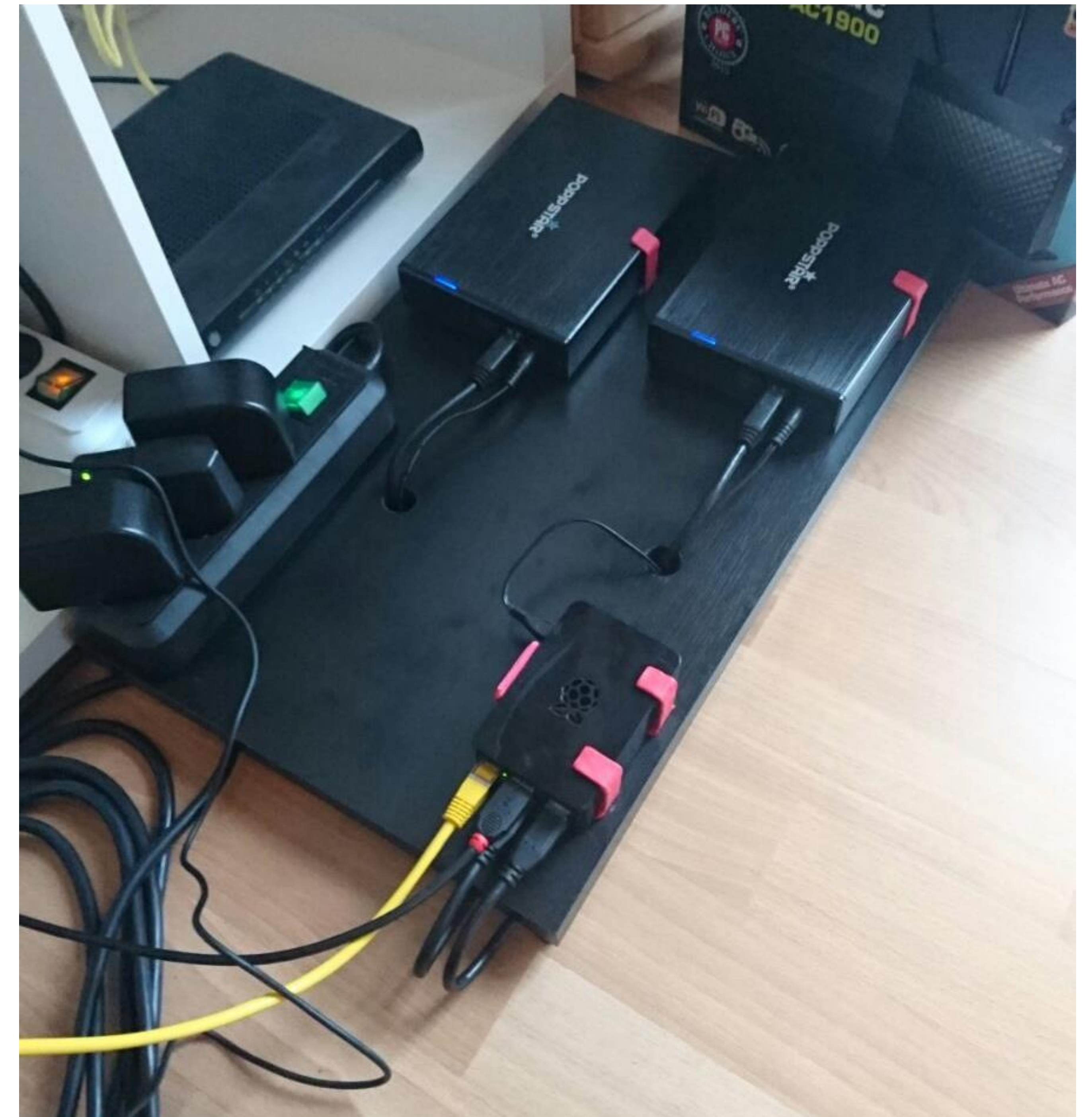
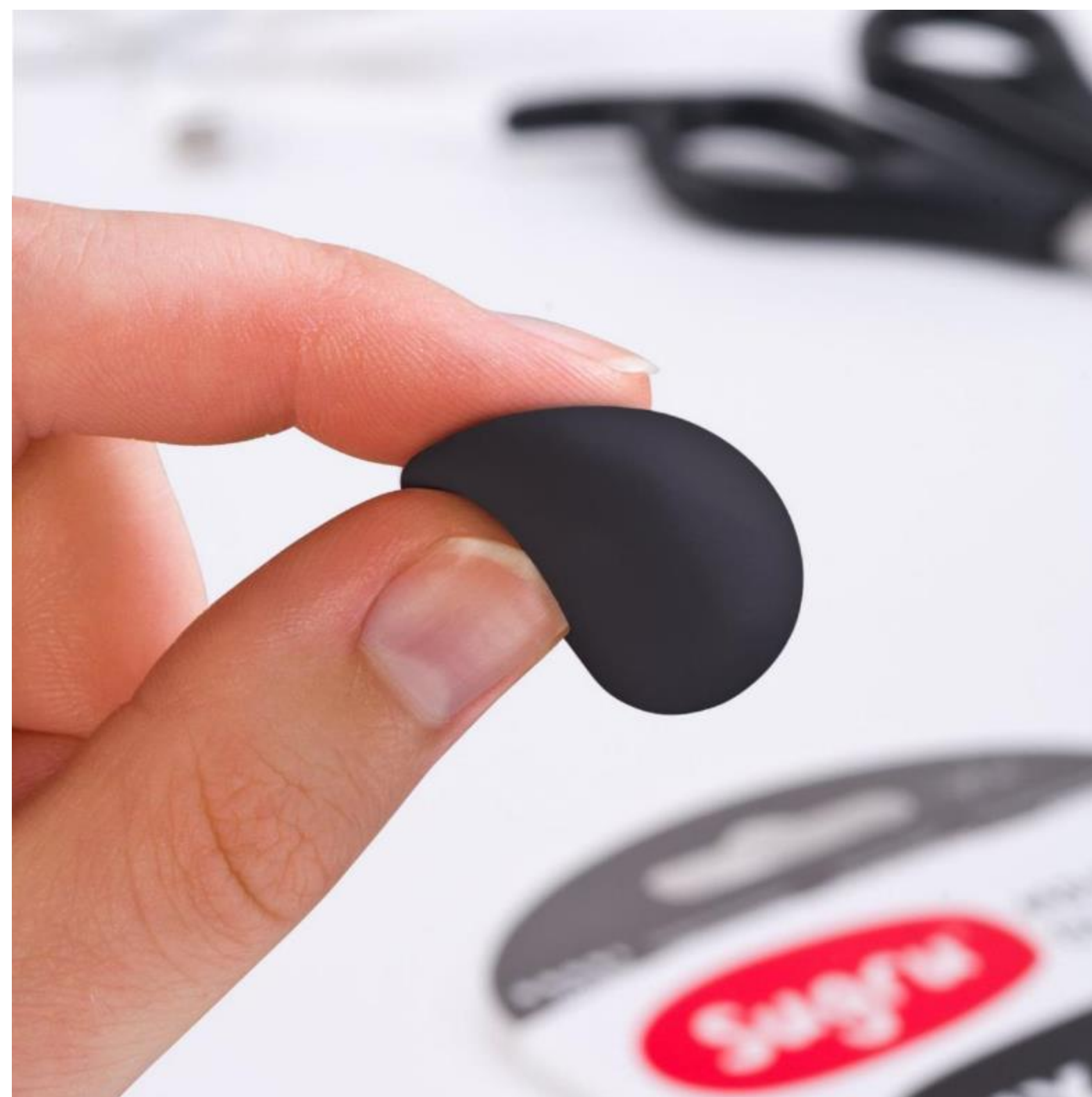
HW Aufbau

- Verkabelung USB / Strom auf Rückseite des Panels mit Kabelschellen



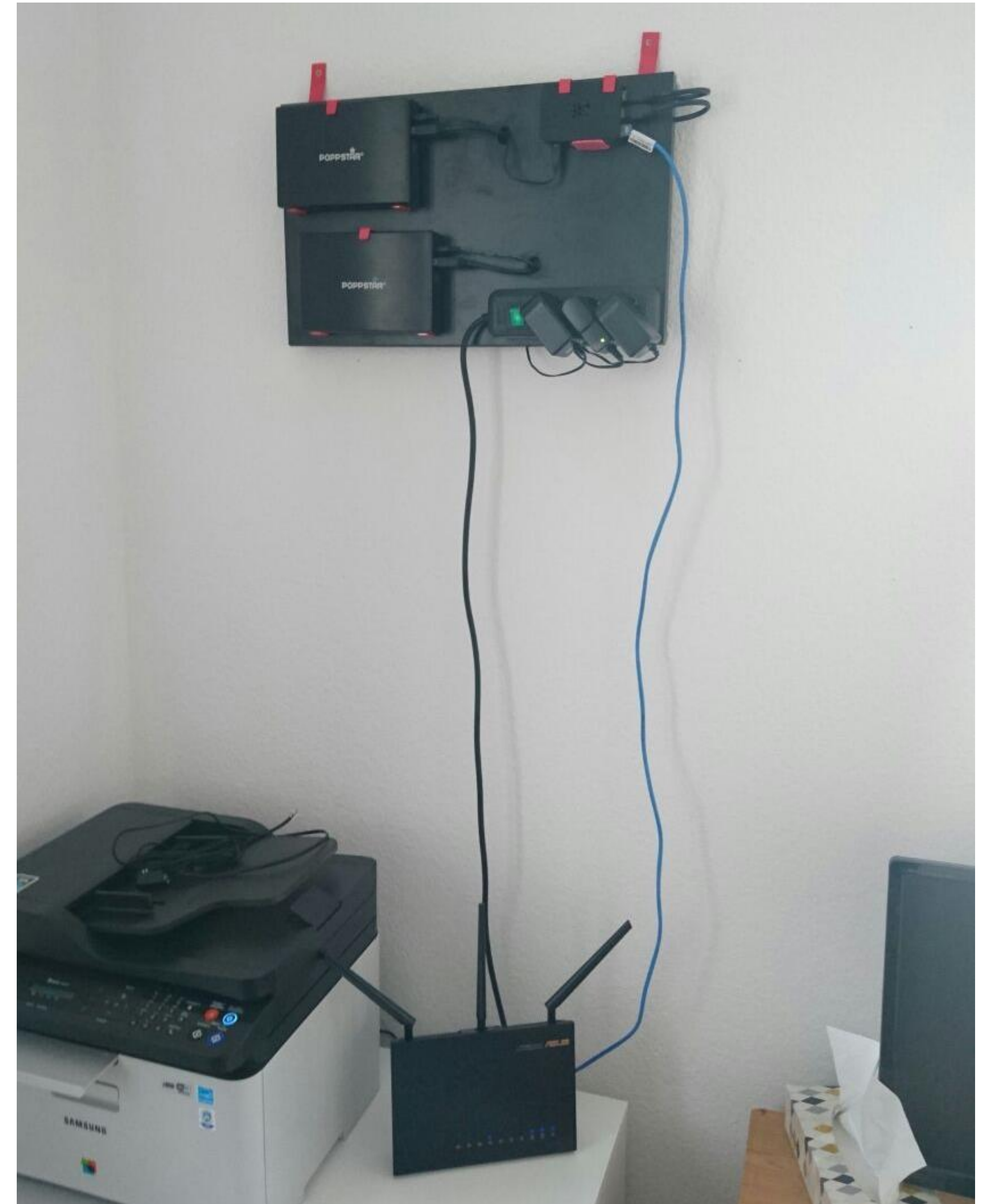
HW Aufbau

- Ränder der Löcher durchs Brett mit Sugru auskleiden



HW Aufbau

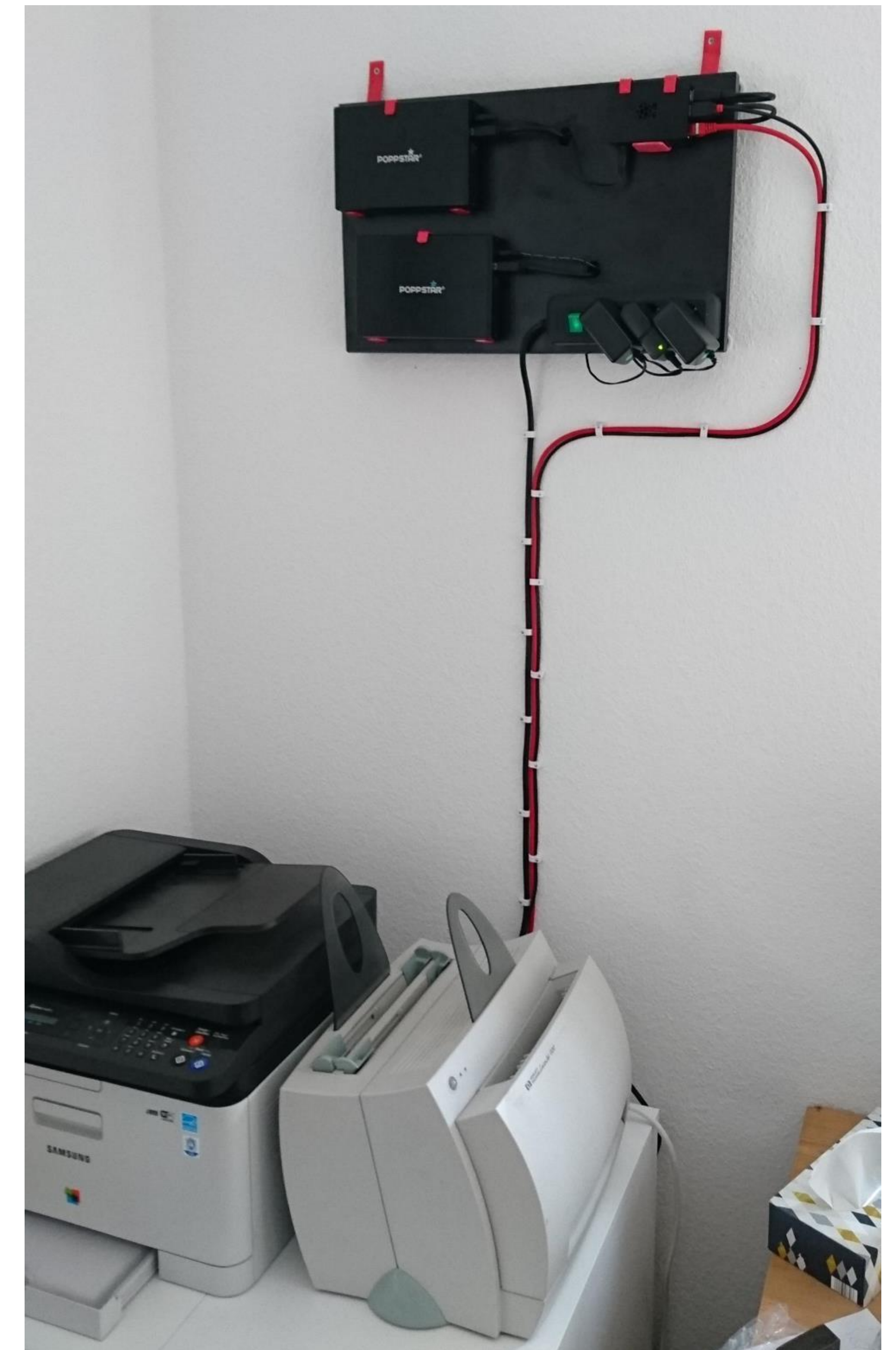
Erstes mal „on the Wall“
→ mit Schrauben in Dübeln
in der Wand befestigt



HW Aufbau

Finale Version

Kabel jetzt mit Kabelschellen geführt
LAN Kabel läuft jetzt über Switch



Raspi2 – NAS-SW

- Grundlegendes zu Linux auf dem Raspberry Pi:
 - Auf dem Pi läuft „Raspbian“, eine Sonderversion von „Debian“ (Linux Distribution) für den Mikrocontroller
 - Raspbian ist eine OpenSource Software
 - Raspbian beinhaltet von Start an
 - einen Webbrowser, meist Firefox oder IceWeasel (abgespeckte Variante von Firefox)
 - Python 2 & Python 3
 - Minecraft
 - LibreOffice
 - Wolfram & Mathematica
 - PDF-Viewer
 - Java IDE
 - nano (extrem simpler Editor, den man häufig verwendet um Einstellungen zu setzen)

Raspi2 – NAS-SW

- Bedienung von Linux
 - Linux wird mit einer „Shell“ (Eingabefenster/Kommandozeile) bedient
 - Wenn man Einstellungen / Dinge starten will, muss oft „sudo“ davorgesetzt werden
„sudo“ steht für „superuser do“
 - Oft verwendeter Befehl in dieser Anleitung: `sudo nano path/myfile`
→ d.h. Superuser fordert nano auf das file „myfile“ im Pfad „path“ zu öffnen

Raspi2 – NAS-SW

- Bedienung von Raspbian

- Updates für den Raspberry Pi werden in der Shell aufgerufen mit:

- ```
sudo apt-get update
```

- Updates werden installiert mit

- ```
sudo apt-get upgrade
```

- Packages sind in einer repository (Sammlung) verfügbar und können automatisiert heruntergeladen werden mit:

- ```
sudo apt-get install XYZ
```

- wobei „XYZ“ ersetzt wird mit dem entsprechenden Paketnamen

- Den Paketnamen kann man googlen oder man erhält auch alle möglichen Pakete über

- ```
sudo apt-cache search Suchbegriff
```


Raspi2 – NAS-SW

- Bevor der Pi als NAS (Network Attached Storage → Netzlaufwerk) agieren kann müssen folgende Dinge erledigt werden:
 1. LAN/WLAN einrichten
 2. VNC muss eingerichtet werden (damit remote Zugriff auf Pi jederzeit möglich ist)
 3. VNC muss nach jedem Neustart automatisch gestartet werden
 4. Festplatten in ext4 Format formatieren (effektiver unter Linux/Raspbian)
 5. Festplatten müssen erkannt und gemountet werden (Zugriff muss von Raspbian aus möglich sein)
 6. Festplatten müssen nach dem Neustart automatisch gemountet werden
 7. Samba Server muss eingerichtet werden (Einfacher Zugriff auf Festplatte von Win-PC aus)
 8. Samba-Server muss automatisch starten
 9. Sicher stellen das Festplatten nicht durchweg laufen, sondern in den stand-by modus gehen können (set hdparm; journaling=off)
 10. Rsync einrichten (alle Daten von einer Festplatte auf eine andere synchronisieren)
→ zunächst testen, dass das einmal funktioniert (alle Daten überprüfen: noch ok?)
 11. Rsync einrichten, dass es jeden Tag (bspw. 5:00 Uhr morgens) automatisch synchronisiert

Raspi2 – NAS-SW 1: LAN Setup



- Beschreibung um das LAN/WLAN einzurichten findet man hier: [Link](#)
- Öffne das Setup-File für die LAN-Adpter (Netzwerk-Anschlüsse):

```
sudo nano /etc/network/interfaces
```

Setze dort

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp
hostname "myRasPi"
# "myRasPi" can be exchanged with an arbitrary name
```

```
allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

- Dann mit „Ctrl+X“ und „Enter“ das Setting-File speichern und den Nano-Editor schließen

Raspi2 – NAS-SW 1: LAN Setup



- Dann müssen die Einstellungen zum Anmelden im LAN bzw. für den Router eingetragen werden. Dazu öffne das File:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

- Und trage dort die Anmeldedaten ein

→ Netzwerkname

→ Netzwerkpasswort

- Verwende „RSN“ Protokoll für ein Netzwerk mit WPA2 (ist üblich)

- Verwende immer CCMP

```
country=DE
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
  ssid="YOURSSID"
  psk="YOURPASSWORD"

  # Protocol type can be: RSN (for WP2) and WPA (for WPA1)
  proto=RSN

  # Key management type
  key_mgmt=WPA-PSK

  # Pairwise can be CCMP or TKIP (for WPA2 or WPA1)
  pairwise=CCMP

  #Authorization option should be OPEN for both WPA1/WPA2
  # (in less commonly used are SHARED and LEAP)
  auth_alg=OPEN
}
```

Nur notwendig, wenn man den
EDIMAX-WLAN-Stick verwendet

Raspi2 – NAS-SW 1: WLAN



- Verhindern dass der EDMIX WLAN-Stick in den sleep-modus geht (andernfalls ist Pi per VNC nicht mehr erreichbar und muss neu gestartet werden):

The problem seems to be that the adapter has power management features enabled by default. This can be checked by running the command:

```
cat /sys/module/8192cu/parameters/rtw_power_mgnt
```

- *A value of 0 means disabled, 1 means min. power management, 2 means max. power management. To disable this, you need to create a new file:*

```
sudo nano /etc/modprobe.d/8192cu.conf
```

- *and add the following:*

```
# Disable power management  
options 8192cu rtw_power_mgnt=0 rtw_enusbss=0
```

- *Once you save the file and reboot your RPi, the WiFi should stay on indefinitely.*

Raspi2 – NAS-SW 2: VNC

Zum nachlesen:

[Link](#)

1. Update durchführen:

```
sudo apt-get update && apt-get upgrade
```

2. Download app „tightvncserver“:

```
sudo apt-get install tightvncserver
```

3. Passwort festlegen:

```
vncpasswd
```

4. Auf Windows PC → TightVNC-Viewer runterladen:

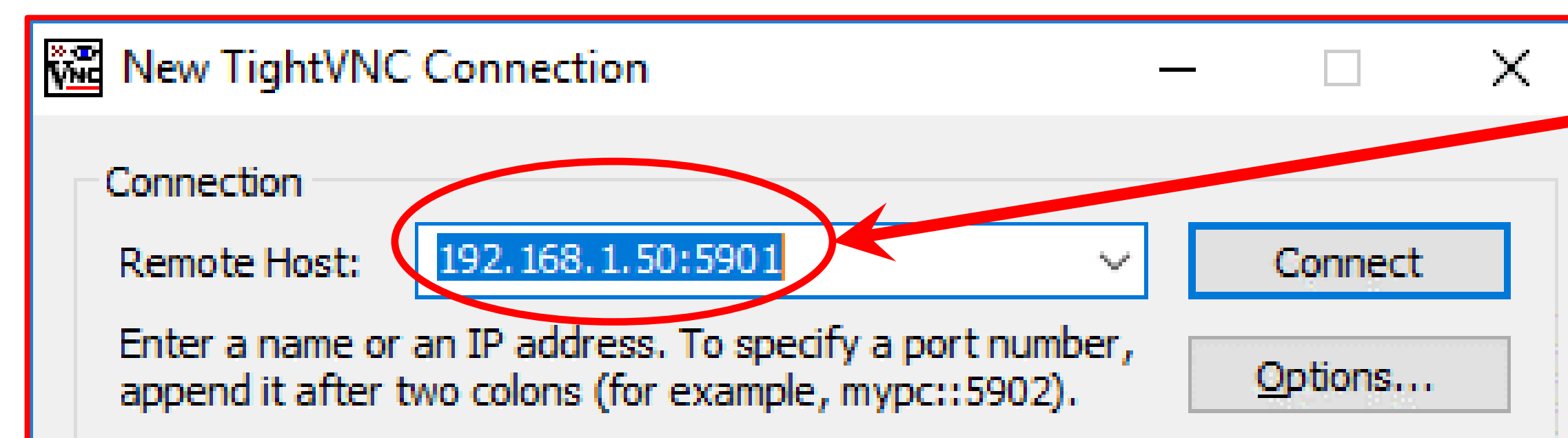
<http://www.tightvnc.com/download.php>

→ nur den Viewer installieren

5. Testen ob eine VNC-Session ohne Autostart möglich ist:

Raspi: `tightvncserver`

Windows:



IP des Raspberry Pi
(idealerweise fix eingetragen
im DHCP-Server des Routers)

Port (hinter „:“) ist immer „5901“

Raspi2 – NAS-SW 2: VNC

Anmerkungen:

- Die Auflösung des VNC kann im init-File angepasst werden indem man die Zeile

```
SU $USER -c '/usr/bin/tightvncserver :1'
```

wie folgt umändert

```
SU $USER -c '/usr/bin/tightvncserver :1 -geometry 1920x1080 -depth 24'
```

es wird empfohlen die Auflösung klein zu halten. Zum Arbeiten geht es trotzdem auf Full-HD. Zum Spielen vermutlich eher schlecht...

- Alternative Dokus:
 - <https://jankarres.de/2012/08/raspberry-pi-vnc-server-installieren/>
 - <https://www.elektronik-kompodium.de/sites/raspberry-pi/2011021.htm>

Raspi2 – NAS-SW 3: VNC Autostart

6. VNC zum Autostart hinzufügen indem folgende Datei erstellt wird:

[Link](#)

Wichtig: Die erste Zeile mit `#!/bin/sh` ist extrem wichtig!!

7. Die Datei kann auch über

wget <http://www.penguintutor.com/otherfiles/tightvncserver-init.txt>

heruntergeladen und über

```
sudo mv tightvncserver-init.txt /etc/init.d/tightvncserver
```

ins richtige Verzeichnis abgelegt werden (muss in init.d liegen!)

Wichtiger Hinweis: Es gibt viele Versionen dieses init-files. Viele haben kein „#!/bin/sh“ in der ersten Zeile. Das ist allerdings absolut wichtig!

8. Zugriffsrechte des Files anpassen:

```
sudo chown root:root /etc/init.d/tightvncserver
```

9. Datei ausführbar machen:

```
sudo chmod 755 /etc/init.d/tightvncserver
```

10. File in die default runlevels hinzufügen (damit es beim start ausgeführt wird)

```
sudo update-rc.d tightvncserver defaults
```

→ **Fertig!**

Raspi2 – NAS-SW 4: ext4 format


1. Formatiere und erstelle Partitionen für beide Festplatten im ext4 Format

Wichtig: „journaling“ soll ausgeschaltet sein

Wichtig: „lazy_itable_init“ muss aus sein

→ das dauert bei der Formatierung zwar lang, aber dann werden nach jedem booten die Festplatten schnell gemountet)

```
sudo mkfs.ext4 -E lazy_itable_init=0,lazy_journal_init=0 /dev/sdXY
```



Anpassen!

2. Um herauszufinden unter welchen Namen die Festplatte im System läuft, geben wir folgenden Befehl ein. In unserem Beispiel ist es /dev/sda

```
sudo fdisk -l
```


Raspi2 – NAS-SW 5: HDD mount

- Zunächst alle nötigen Apps installieren

```
sudo apt-get -y install ntfs-3g hfsutils hfsprogs exfat-fuse
```

- Das Tool ntfs-3g ist notwendig um das Windows-Format „NTFS“ in Linux lesen zu können
- Erstelle einen Ordner in den wird die Festplatte mounten können (d.h. in diesem Ordner wird angezeigt was auf der Festplatte ist)

```
sudo mkdir /media/usbstick
```

- Liste alle Datenträger auf, die gerade angeschlossen sind:

```
sudo blkid -o list -w /dev/null                oder                sudo lsblk
```

- USB-Sticks heißen meist *sda* und externe Festplatten *hd*, ggf. mit anhängender Nummer. Uns interessiert die erste und letzte Spalte des richtigen Gerätes mit dessen *device* Pfad und *UUID*.

Raspi2 – NAS-SW 5: HDD mount

- Mounten eines Laufwerks mit Datentyp...

- FAT32

- ```
sudo mount -t vfat -o utf8,uid=pi,gid=pi,noatime /dev/sda /media/usbstick
```

- NTFS

- ```
sudo mount -t ntfs-3g -o utf8,uid=pi,gid=pi,noatime /dev/sda /media/usbstick
```

- HFS+

- ```
sudo mount -t hfsplus -o utf8,uid=pi,gid=pi,noatime /dev/sda /media/usbstick
```

- exFAT

- ```
sudo mount -t exfat -o utf8,uid=pi,gid=pi,noatime /dev/sda /media/usbstick
```

- ext4

- ```
sudo mount -t ext4 -o defaults /dev/sda /media/usbstick
```

- Um ein Laufwerk zu unmounten (herauszunehmen)

- ```
sudo umount /media/usbstick
```


Raspi2 – NAS-SW 6: Auto HDD mount

- Soll der USB-Stick oder die USB-Festplatte nun bei jedem Start des Raspberry Pis automatisch eingebunden werden, können wir dies für einen Eintrag in *fstab* erlauben, wozu wir die zuvor ausgelesene UUID benötigen. *fstab* können wir editieren:

```
sudo nano -w /etc/fstab
```

- Die UUID bekommt man über die Liste mit:

```
sudo blkid -o list -w /dev/null
```

Raspi2 – NAS-SW 6: Auto HDD mount

- Ergebnis für

```
sudo blkid -o list -w /dev/null
```

ist:

- Interner Speicher
- NAS-Festplatte (Backup)
- NAS Festplatte

```
pi@raspberrypi ~ $ sudo blkid -o list -w /dev/null
```

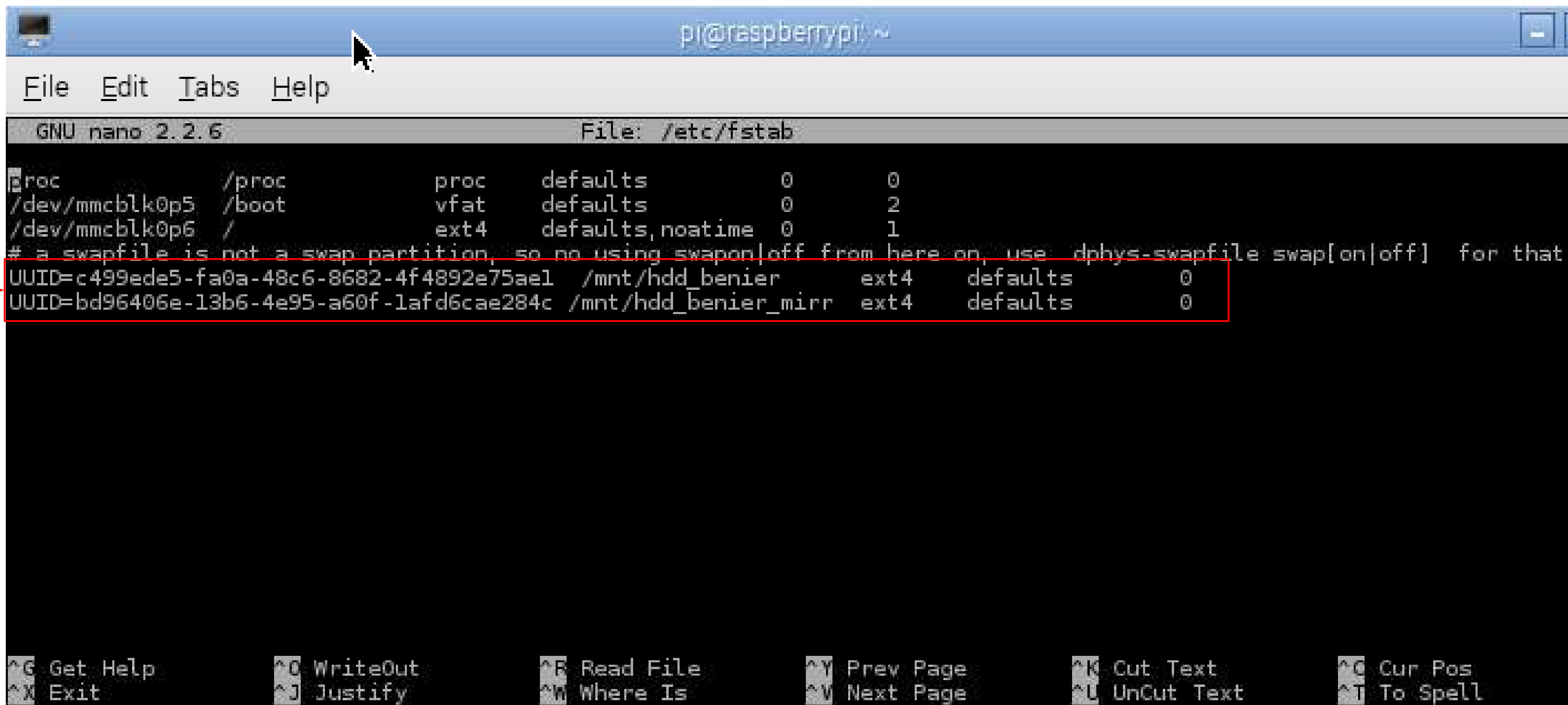
device	fs_type	label	mount point	UUID
/dev/mmcblk0p1	vfat	RECOVERY	(not mounted)	C45A-8763
/dev/mmcblk0p3	ext4	SETTINGS	/media/SETTINGS__	74113372-a9d6-452c-b135-a528a2182b3c
/dev/mmcblk0p5	vfat	boot	/boot	0149-FE18
/dev/mmcblk0p6	ext4	root	/	cc1392cf-0c3e-441f-abe1-964ef4aad79d
/dev/sdb1	ext4	hdd_benier_mirr	/mnt/hdd_benier_mirr	bd96406e-13b6-4e95-a60f-1afd6cae284c
/dev/sda1	ext4	hdd_benier	/mnt/hdd_benier	c499ede5-fa0a-48c6-8682-4f4892e75ae1

Jeweilige UUID
zum Datengträger

Raspi2 – NAS-SW 6: Auto HDD mount

- In `/etc/fstab` eintragen welche Festplatte mit welcher UUID beim hochfahren wohin gemounted werden soll. Bsp für *fstab*:

- Hinzugefügte Befehle zu `fstab` um die Festplatten zu booten



```
pi@raspberrypi:~  
File Edit Tabs Help  
GNU nano 2.2.6 File: /etc/fstab  
proc /proc proc defaults 0 0  
/dev/mmcblk0p5 /boot vfat defaults 0 2  
/dev/mmcblk0p6 / ext4 defaults,noatime 0 1  
# a swapfile is not a swap partition, so no using swapon/off from here on, use dphys-swapfile swap[on|off] for that  
UUID=c499ede5-fa0a-48c6-8682-4f4892e75ae1 /mnt/hdd_benier ext4 defaults 0  
UUID=bd96406e-13b6-4e95-a60f-1afd6cae284c /mnt/hdd_benier_mirr ext4 defaults 0  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^_ To Spell
```

Raspi2 – NAS-SW 6: Auto HDD mount

- Alternative Dokus dazu:
 - <https://jankarres.de/2013/01/raspberry-pi-usb-stick-und-usb-festplatte-einbinden/>
 - <http://raspberrypi.stackexchange.com/questions/10550/how-to-connect-raspberry-pi-to-a-western-digital-mybooklive-nas-and-create-a-link>

Raspi2 – NAS-SW 7&8: Samba server

- Samba Server installieren

```
sudo apt-get install samba samba-common-bin
```

- Bearbeite die Konfiguration

```
sudo nano /etc/samba/smb.conf
```

und ersetze in „smb.conf“

```
# security = user
```

mit

```
security = user
```

(nicht mehr auskommentiert lassen, da wir ein Passwort setzen wollen!)

- Jetzt lege einen Samba Nutzer „pi“ an und lege ein Passwort fest
(Das ist das Passwort das man später im Windows eingibt!)

```
sudo smbpasswd -a pi
```

Raspi2 – NAS-SW 7&8: Samba server

- Weise dem Samba Server einen Datenträger (Festplatte / USB-Stick) zu:

```
sudo chown -R pi:pi /media/usbstick
```

- Definiere in der smb.conf wie auf den Datenträger zugegriffen werden darf (Nur lesen/schreiben? Immer mit Passwort?). Also rufe

```
sudo nano /etc/samba/smb.conf
```

Und trage ganz unten in „smb.conf“ ein:

```
[MeinLaufwerk]
path = /media/usbstick
writeable=yes
guest ok=no
workgroup=WORKGROUP
browseable=yes
```

Das wird später im Windows
als Laufwerksname angezeigt

Dadurch wird das Laufwerk
im Windows beschreibbar und lesbar

Alle die darauf zugreifen wollen
müssen das Passwort wissen

Anmeldung in der Windows-
Netzwerkgruppe „WORKGROUP“
→ besserer Windows Zugriff

Schnellerer Zugriff wenn man Bilder
über den Samba Server anschaut

Raspi2 – NAS-SW 7&8: Samba server

- Sobald alles eingestellt ist, starte den Samba Server neu:

```
sudo /etc/init.d/samba restart
```

und teste, ob der Zugriff von Windows aus funktioniert.

Raspi2 – NAS-SW 7&8: Samba server

- Alternative Dokus dazu:
 - <https://jankarres.de/2013/11/raspberry-pi-samba-server-installieren/>

Raspi2 – NAS-SW 9: HDD standby

1. RasPi updaten und „hdparm“ installieren:

```
sudo apt-get update  
sudo apt-get -y install hdparm
```

2. Um herauszufinden unter welchen Namen die Festplatte im System läuft, geben wir folgenden Befehl ein. In unserem Beispiel ist es /dev/sda

```
sudo fdisk -l
```

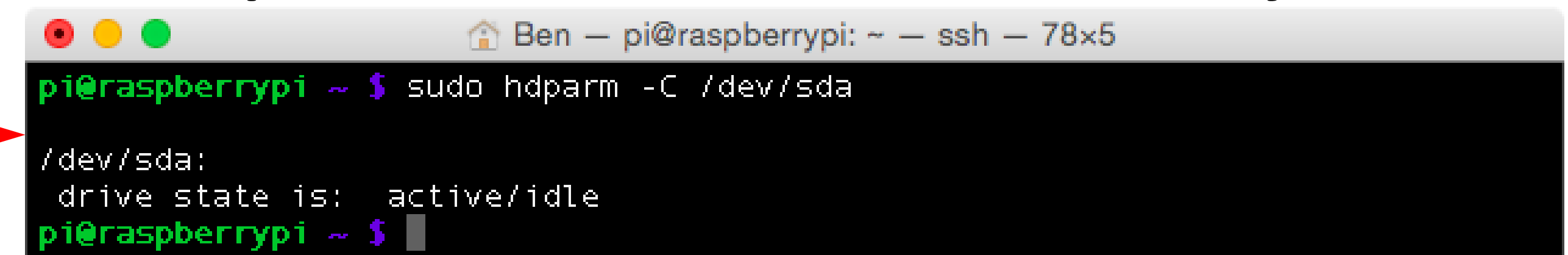


```
Ben — pi@raspberrypi: ~ — ssh — 78x23  
pi@raspberrypi ~ $ sudo fdisk -l  
  
Disk /dev/mmcblk0: 15.9 GB, 15931539456 bytes  
4 heads, 16 sectors/track, 486192 cylinders, total 31116288 sectors  
Units = sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disk identifier: 0x00092fac  
  
    Device Boot      Start         End      Blocks   Id  System  
/dev/mmcblk0p1        8192       122879        57344    c   W95 FAT32 (LBA)  
/dev/mmcblk0p2     122880     31116287     15496704    83   Linux  
  
Disk /dev/sda: 160.0 GB, 160041885696 bytes  
81 heads, 63 sectors/track, 61254 cylinders, total 312581808 sectors  
Units = sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disk identifier: 0xc8cd1456  
  
    Device Boot      Start         End      Blocks   Id  System  
/dev/sda1           2048     312581807     156289880    83   Linux  
pi@raspberrypi ~ $
```

Raspi2 – NAS-SW 9: HDD standby

3. Überprüfe den aktuellen Power-Mode-Status. Falls du den Status *unknown* zurück bekommst lässt sich die Festplatte nicht in Standby versetzen.

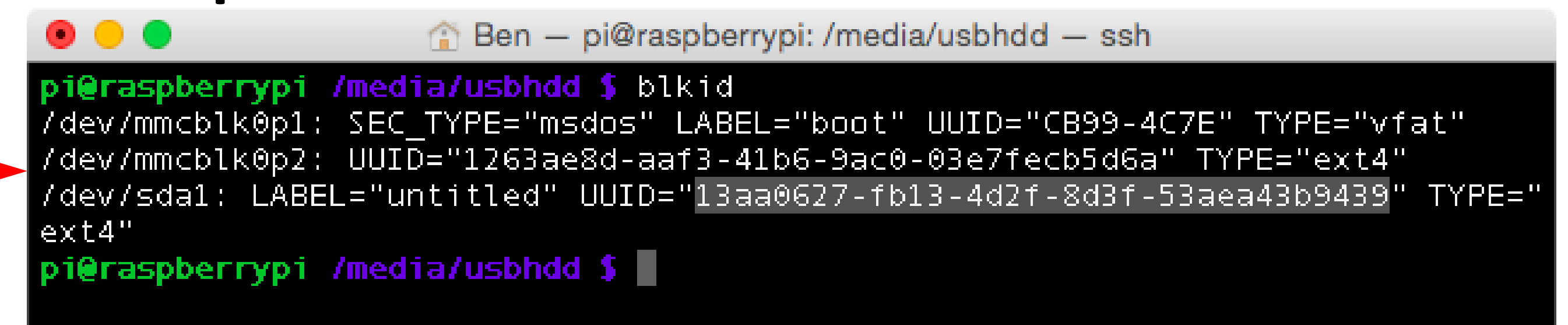
```
sudo hdparm -C /dev/sda
```



```
Ben — pi@raspberrypi: ~ — ssh — 78x5
pi@raspberrypi ~ $ sudo hdparm -C /dev/sda
/dev/sda:
drive state is: active/idle
pi@raspberrypi ~ $
```

4. Wir benötigen die UUID der Festplatte. Gib folgenden Befehl in die Console ein und notier dir die UUID der Festplatte.

```
sudo blkid
```



```
Ben — pi@raspberrypi: /media/usbhdd — ssh
pi@raspberrypi /media/usbhdd $ blkid
/dev/mmcblk0p1: SEC_TYPE="msdos" LABEL="boot" UUID="CB99-4C7E" TYPE="vfat"
/dev/mmcblk0p2: UUID="1263ae8d-aaf3-41b6-9ac0-03e7fecb5d6a" TYPE="ext4"
/dev/sda1: LABEL="untitled" UUID="13aa0627-fb13-4d2f-8d3f-53aea43b9439" TYPE="ext4"
pi@raspberrypi /media/usbhdd $
```

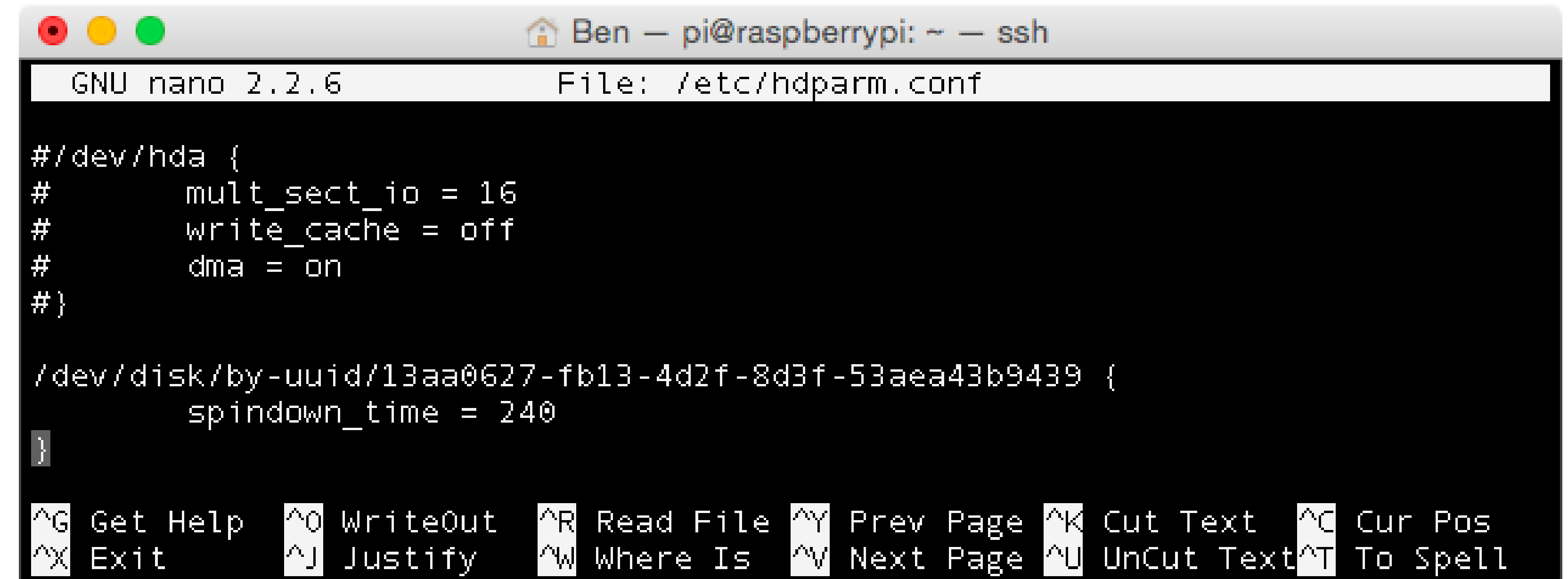
5. Öffne die hdparm config Datei

```
sudo nano /etc/hdparm.conf
```

Raspi2 – NAS-SW 9: HDD standby

6. Im File „/etc/hdparm.conf“ setze:

7. Deaktiviere das journaling der Festplatte (damit sie in den Standby gehen kann):



```
GNU nano 2.2.6 File: /etc/hdparm.conf

#/dev/hda {
#     mult_sect_io = 16
#     write_cache = off
#     dma = on
#}

/dev/disk/by-uuid/13aa0627-fb13-4d2f-8d3f-53aea43b9439 {
    spindown_time = 240
}

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```


Raspi2 – NAS-SW 9: HDD standby

7. Deaktiviere das journaling der Festplatte (damit sie in den Standby gehen kann):

```
sudo -i                                # (gehe in root User)
# alle Dienste, die allergisch auf fehlenden Dateisystemzugriff allergisch reagieren
# herunterfahren (sshd natuerlich anlassen), z.B.:
service mysql stop
service postfix stop
service dovecot stop
service rsyslog stop
# usw...

# Magie! Alle Partitionen aushaengen und read-only remounten
echo u > /proc/sysrq-trigger

# Cache auf Platte schreiben
echo s > /proc/sysrq-trigger

# Journaling abschalten: das ^ steht fuer deaktivieren
tune2fs -O ^has_journal /dev/mmcblk0p2

# ein nachfolgender fsck ist wichtig.
e2fsck -fy /dev/mmcblk0p2

# und wieder Cache auf Platte schreiben
echo s > /proc/sysrq-trigger

# und sofortiger Neustart ohne irgendwelche Schreibzugriffe
echo b > /proc/sysrq-trigger
```

Raspi2 – NAS-SW 10: rsync

- Rsync ist ein Programm mit dem zu vorgegebenen Zeiten die Daten von einem Datenträger auf einen anderen synchronisiert werden
= Erstellung eines Backups/Mirrors
- Die Synchronisation funktioniert als shell Befehl direkt oder automatisiert zu einer bestimmten Uhrzeit oder sogar an einem bestimmten Tag/Woche
- Im folgenden Beispiel existieren zwei Festplatten, die gemountet sind in
 1. /media/usbstick
 2. /media/usbstick_mirr

Raspi2 – NAS-SW 10: rsync

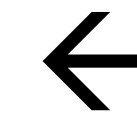
1. rsync installieren

```
sudo apt-get update  
sudo hdparm -C /dev/sda
```

2. rsync an einem File testen

→ funktioniert der Kopier-Prozess überhaupt?

```
rsync -av --delete /media/usbstick/ /media/usbstick_mirr/
```



Die 2. Festplatte muss hierzu schon gemountet sein damit du Tests durchführen kannst

– dabei setzt sich die Option `-av` aus folgenden Befehlen zusammen:

`-v` zeigt während des Synchronisierens alle ausgeführten Schritte in der Shell an
`-a` fasst folgende Optionen zusammen:

- `-r` kopiert Unterverzeichnisse
- `-l` kopiert symbolische Links
- `-p` behält Rechte der Quelldatei bei
- `-t` behält Zeiten der Quelldatei bei,
- `-g` behält Gruppenrechte der Quelldatei bei

– die Option `--delete` vergleicht Quellverzeichnisse und Zielverzeichnisse und sorgt dafür, dass Dateien, die im Quellverzeichnis nicht (mehr) vorhanden sind, im Zielverzeichnis gelöscht werden

Raspi2 – NAS-SW 10: rsync

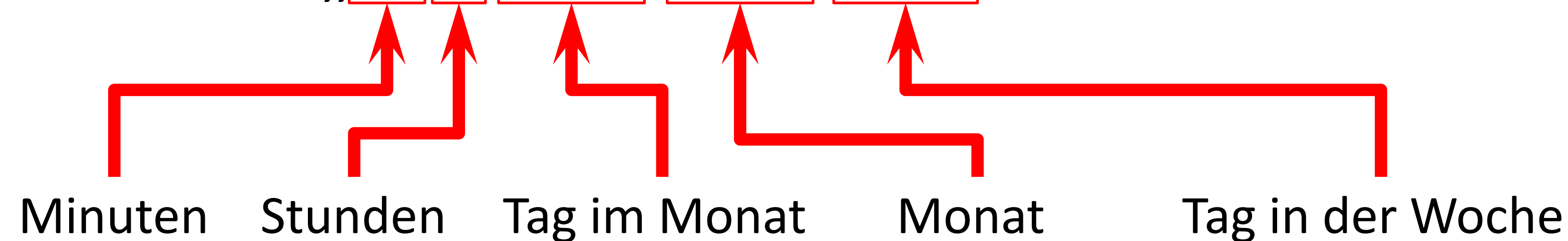
3. Automatisiert die Synchronisierung einrichten. Dazu erstmal einen cron tab starten:

```
sudo crontab -e
```

Dieser Befehl öffnet ein config-File für rsync im Nano-Editor

(Achtung: Unbedingt mit „sudo“! Ohne geht zwar auch, aber er öffnet das falsche File!!)

4. Im crontab config-File steht der selbe „rsync ...“-Befehl wie vorhin, allerdings mit einem Präfix, der sich aus „**m****h** **dom** **mon** **dow** *command*“ zusammensetzt




alle angaben die nicht benötigt werden, werden durch ein „*“ ersetzt. D.h. ein Backup jeden Tag um 5 Uhr morgens für unsere Festplatten wird erledigt mit:

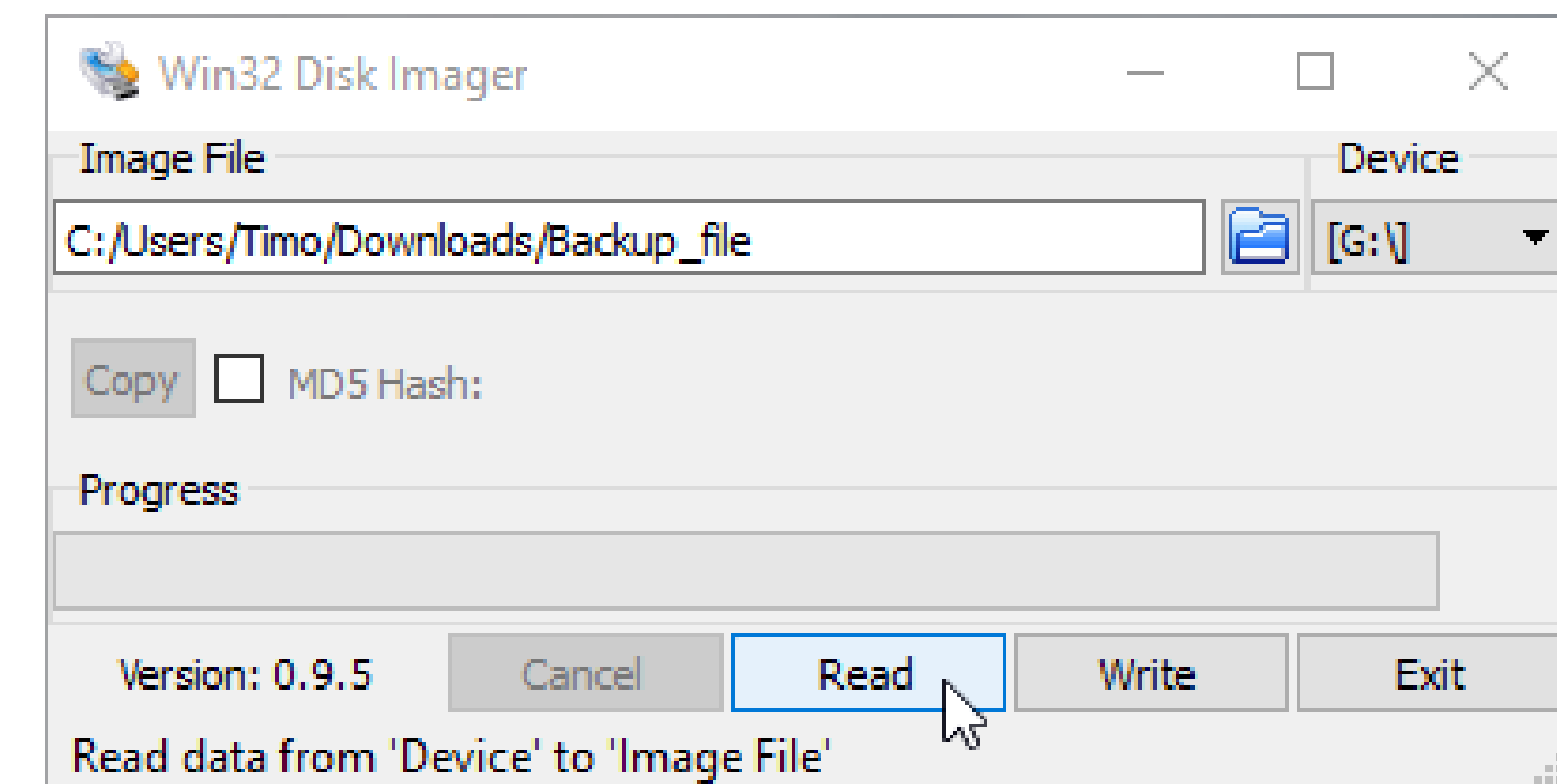
```
00 05 * * * rsync -av --delete /media/usbstick/ /media/usbstick_mirr
```

Raspi2 – NAS-SW 10: rsync

- Alternative Dokus:
 - <https://www.howtogeek.com/139433/how-to-turn-a-raspberry-pi-into-a-low-power-network-storage-device/>
(Allgemeine Beschreibung wie man alles einrichtet)
 - <https://wiki.ubuntuusers.de/rsync/>
(Bedienungsanleitung für das Modul „rsync“)

Raspi2 – System backups

- Hier wird beschrieben wie du ein Backup von der SD-Karte im Raspberry Pi mit dem System und allen Einstellungen machen kannst
Backup: SD-Karten-Inhalt speichern auf Windows-PC als Image file
1. Am Windows PC: Lade das Programm „Win32DiskImager“ herunter
 2. Entnehme die SD-Karte aus dem Raspberry Pi und schließe sich am Windows-PC an und Starte den Win32DiskImager
 3. Wähle das Laufwerk mit der SD-Karte aus und definiere über  mit welchem Dateinamen in welchem Ordner das Backup-file abgelegt werden soll
 4. Erstelle das Image-File mit „Read“
Achtung: Der komplette Inhalt der SD-Karte wird in ein Imagefile gesteckt.
D.h. von einer SD-Karte mit 32GB Speicherplatz wird ein 32GB Imagefile erstellt.



Raspi2 – Druckerserver

- Damit Pi als Druckerserver agieren kann muss CUPS installiert werden
- CUPS = **C**ommon **U**nix **P**rinting **S**ystem
- Das CUPS ist mit dem Browser auf dem Port 631 mit der IP des Raspberry Pi (Bsp. 192.168.1.10) erreichbar, z.B. 192.168.1.10:631
- Wenn CUPS von Windows PCs benutzt wird, ist es nötig einen [Samba Server](#) vorher auf dem Raspberry Pi einzurichten
- CUPS zeigt im Browser (192.168.1.10:631) auch die aktuellen Druckaufträge an

Raspi2 – Druckerserver

1. Installiere CUPS und gutenprint Druckertreiber (hat fast alle nötigen Treiber)

```
sudo apt-get install cups printer-driver-gutenprint
```

2. Öffne die Samba Server config

```
sudo nano /etc/samba/smb.conf
```

im Bereich „====Share Definitions====“ gibt es einen Abschnitt „[printers]“ der noch auskommentiert ist. Kommentiere diese Stelle ein und passe sie wie folgt an:

```
[printers]
comment = All Printers
browseable = no
path = /var/spool/samba
printable = yes
guest ok = no
read only = yes
create mask = 0700
```

Raspi2 – Druckerserver

3. Einen Admin einrichten für die Bedienung des CUPS Systems auf dem Browser:

```
sudo usermod -aG lpadmin pi
```

→ Das CUPS Passwort ist jetzt das selbe Passwort des Admin users auf dem Pi

4. Das Passwort kann geändert werden mit:

```
sudo passwd pi
```


Raspi2 – Druckerserver

5. Einen neuen Drucker kann man im CUPS System einfach selbst hinzufügen unter „Verwaltung“ → „Drucker hinzufügen“:



Evtl. verlangt der Pi, dass man sich als Admin einloggt...