

# Rund um Technik

Auf dieser Seite trage ich technische Informationen zu alten und neueren Geräten zusammen um sie für mich und andere (Bastler) zu erhalten... 😊 .

[Gartengeräte](#) aller Preisklassen, deren baugleiche Verwandte, Reparaturen, Tipps und Tricks 😊

[Kommunikationsgeräte](#) wie z.B. VoIP Adapter etc. 😊

## Temperatur & Feuchte Datenlogger TFD 128

Seit einiger Zeit gesitze ich den [TFD 128 von ELV](#). Die mitgelieferte Software läuft jedoch nur unter Microsoft Windows so dass ich mich nach einer Alternative für Linux umgesehen habe. Folgende Quellen sind dabei sehr nützlich:

- <http://projects.nesrada.de/tfd128/>
- <http://www.netzmafia.de/skripten/hardware/TFD128/>
- <http://vd-software.inside1.net/TFD128/>

Die letzte Quelle ist leider wieder eine reine Windows Software, bietet dafür aber einen deutlich größeren Funktionsumfang als die Software von ELV.

## Canon EOS 600D

### Vorbereitung

```
# Tool installieren
sudo apt-get install gphoto2
# Kamera erkenne/checken
gphoto2 --list-cameras
```

### Reihenaufnahme

```
# Intervallaufnahme starten, hier alle 10 Sekunden ein Bild, 300 Bilder
insgesamt
gphoto2 --capture-image-and-download --filename=%H_%M_%S_%.%C --interval 10 -
F 300
```

# Minolta Dimage 7Hi

## Speicherkarte

Die Minolta Dimage 7Hi ist empfindlich was die Größe, die Partitionierung und Formatierung der CompactFlash Karte an geht. Sie verträgt max. 2GB große CF-Karten die mit einer primären Partition vom Typ 0x06 (FAT16) und mit FAT16 formatiert sein müssen. Anderenfalls wird die Karte nicht erkannt und es können keine Bilder gespeichert werden.

Die Formatierung einer CF-Karte mit Hilfe der Kamera-Firmware ist nicht möglich daher muss eine „neue“ CF-Karte am PC vorbereitet werden. Sehr gute Dienste leistet das Linux Tool GParted [siehe hier](#). Es steht für nahezu jede Linux Distribution über die Paketquellenverwaltung zur Verfügung.

## Netzteil

Die Minolta Dimage 7Hi verfügt über eine Buchse für Hohlstecker zur Spannungsversorgung über ein Steckernetzteil.

Da die Kamera einen [Einschaltstrom](#) von ca. 1,5A benötigt sollte das zu verwendende Netzteil etwas leistungstärker ausgelegt sein. Ich verwende eines mit einem maximalen Ausgangsstrom von 2A bei 6V DC, das hat bisher immer ausgereicht.

Passende Stecker findet man z.B. bei Reichelt unter folgenden Bezeichnungen:

Bezeichnung	Artikel-Nr.:	Preis:
DC-Stecker 1,7mm / 4,0mm / 9,5mm	<a href="#">HS 17-10</a>	0,84 EUR
Hohlstecker, gerade, Øi=1,7mm Øa=4,0mm, JS	<a href="#">LUM 1636-02</a>	0,92 EUR
DC-Winkelstecker 1,7mm / 4,0mm / 9,5mm	<a href="#">HSW 17-10</a>	1,15 EUR

**HINWEIS:** Steckerbelegung: innen „+“ (PLUS), außen „-“ (MINUS)

## Zeitraffer erstellen

Meine alte Minolta Dimage 7Hi lässt sich noch prima für Intervallaufnahmen verwenden. Hat man erst einmal z.B. über einen Zeitraum von 6 Stunden jede Minute ein Bild geschossen so kann man daraus einen kleinen Zeitraffer-Film erstellen.

[make\\_movie.sh](#)

```
#!/bin/bash
# Kleines Skript zum Erstellen eines Zeitraffer-Films aus einem Haufen
# Einzelbilder.
#
echo -n "Konvertiere Dateien ..."
for datei in PIC*.JPG
do
#  convert -resize x1080 -quality 50 "$datei" "small_$datei"
  convert -resize x1080 "$datei" "small_$datei"
  #rm $datei
```

```
done
echo " done"

echo -n "renaming ..."
LENGTH=`ls | wc -l | wc -L`
NUMBER="1"
find small_*.JPG | while read foto
do
    mv $foto "`printf %0${LENGTH}d $NUMBER`".jpg
    NUMBER=$((NUMBER + 1))
done
echo " done"

echo -n "Film erstellen ..."
# ffmpeg -f image2 -i %0${LENGTH}d.jpg -r 25 -vcodec libx264 output.mpg
# ffmpeg -f image2 -i %0${LENGTH}d.jpg -r 25 -vb 4096k output.mpg
# mencoder mf://`date +%F`.jpg -mf fps=20:type=jpg -o video.flv -of
lavf -ovc lavc -oac lavc -lavcopts vcodec=flv:vbitrate=1000 -vf
crop=640:346:0:0
mencoder mf://%0${LENGTH}d.jpg -mf fps=15:type=jpg -o video.flv -of
lavf -ovc lavc -oac lavc -lavcopts vcodec=flv:vbitrate=2000 -vf
crop=1920:1080:0:0
echo " done"
```

Nicht vergessen make\_movie.sh mit `chmod +x make_movie.sh` auch ausführbar zu machen



## Zeitraffer erstellen - die Zweite

Voraussetzung: ein Verzeichnis mit vielen Fotos - Dateiname nach Schema: nnnn.jpg

```
ls *.jpg > files.txt
mencoder -nosound -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=21600000 -o
Gartenhaus_1920x1080.avi -mf type=jpeg:fps=24 mf://@files.txt -vf
scale=1920:1080
mencoder -nosound -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=21600000 -o
Gartenhaus_1440x1080.avi -mf type=jpeg:fps=24 mf://@files.txt -vf
scale=1440:1080
```

## Zeitraffer erstellen - die Dritte

Voraussetzung: ein Verzeichnis mit vielen Fotos

```
# Bilder vorher zuschneiden:
mogrify -resize 1920x1080 *.jpg
# ansonsten direkt zusammenschneiden...
ffmpeg -r 30 -pattern_type glob -i "*.jpg" -c:v libx264 -pix_fmt yuv420p -
```

```
movflags +faststart timelapse.mp4
```

## Edimax IC-3110W Webcam mit WLAN

### Snapshots aufzeichnen

```
wget --user=[BENUTZERNAME] --password=[PASSWORT]  
http://[IP-ADRESSE]/jpg/image.jpg
```

### Livestream anschauen

```
vlc rtsp://[IP-ADRESSE]:554/ipcam.sdp
```

oder für H264

```
vlc rtsp://ipcam:554/ipcam_h264.sdp
```

## Pimp my Synology DS214

### Samba

Wem die Konfigurationsmöglichkeiten der Synology Weboberfläche nicht ausreichen kann z.B. die Samba Konfiguration per Kommandozeile (SSH Login auf der Synology DS214 vorausgesetzt) erweitern. Dazu einfach `vi /usr/syno/etc/smb.conf` eingeben und schon steht einem die große

Welt der Samba-Konfiguration offen 😊

Einen Neustart des Samba Servers erreicht man anschließend mit:

```
./usr/syno/etc/rc.sysv/S80samba.sh restart
```

Meine aktuelle `smb.conf` sieht folgendermaßen aus:

```
[global]  
    printcap name=cups  
    winbind enum groups=yes  
    unix extensions=yes  
    security=user  
    local master=no  
    realm=*  
    passdb backend=smbpasswd  
    printing=cups  
    max protocol=SMB2  
    winbind enum users=yes  
    load printers=yes  
    workgroup=wg  
[home]
```

```
case sensitive = yes
follow symlinks = yes
wide links = yes
create mode = 0777
comment = home
browseable = no
writable = yes
valid users = %U
directory mode = 0777
preserve case = yes
path = /var/services/homes/%U
case sensitive=yes
    preserve case=yes
follow symlinks = yes
wide links = yes
```

[DS214]

```
invalid users=nobody,nobody
valid users=nobody,nobody
comment=""
path=/volume1/DS214
guest ok=yes
browseable=yes
fileindex=no
mediaindex=no
edit synoacl=yes
win share=yes
skip smb perm=yes
enable recycle bin=yes
recycle bin admin only=yes
hide unreadable=yes
ftp disable list=yes
ftp disable modify=yes
ftp disable download=yes
read list=nobody,nobody
write list=nobody,nobody
writeable=yes
```

[homes]

```
invalid users=nobody,nobody
valid users=nobody,nobody
comment="user home"
path=/volume1/homes
guest ok=yes
browseable=yes
fileindex=no
mediaindex=yes
edit synoacl=yes
win share=yes
skip smb perm=yes
enable recycle bin=yes
recycle bin admin only=no
hide unreadable=yes
```

```
ftp disable list=no
ftp disable modify=no
ftp disable download=no
read list=nobody,nobody
write list=nobody,nobody
writeable=yes
case sensitive=yes
preserve case=yes
follow symlinks = yes
wide links = yes
```

Folgende Zeilen habe ich zur Standard-Konfiguration hinzugefügt:

case sensitive=yes - Groß-/Kleinschreibung von Dateinamen beachten

preserve case=yes - Schreibweise beibehalten, die der Client liefert

follow symlinks = yes- symbolische Links verfolgen

wide links = yes - symbolische Links außerhalb der Freigabe verfolgen

Quelle: <http://www.oreilly.de/german/freebooks/samba2ger/ch08.html>

## rsync

Zuerst den Netzwerksicherungsdienst auf dem Synology NAS aktivieren. Die Option findet sich unter Datensicherung & Replikation > Sicherungsdienste > Netzwerksicherungsziel

```
rsync -a --timeout=600 tmp/Backup
christoph@192.168.100.10:/volume1/homes/christoph/test/
```

Quelle: <https://www.synology.com/de-de/knowledgebase/faq/372>

From:

<https://von-thuelen.de/> - **Christophs DokuWiki**

Permanent link:

<https://von-thuelen.de/doku.php/wiki/technik/uebersicht?rev=1655050822>

Last update: **2022/06/12 18:20**

