

Tutorial: Raspberry Pi 8MP Überwachungskamera mit Motion

Tutorial: Raspberry Pi als 8 Megapixel Webcam/Überwachungskamera mit Motion-Detection.

Installation und Konfiguration des Softwarepaketes Motion aufdem Raspberry Pi

In diesem Projekt-Tutorial zeige ich Euch, wie Ihr die 8 Megapixel Raspi-Cam mit dem Raspberry Pi verbinden könnt und welche Einstellungen Ihr dafür am Pi vornehmen müsst, dass sie auch funktioniert.

Das Software-Paket Motion bietet nicht nur einen eigenen Webserver zur Ansicht und das Streaming von Kamerabildern an, Motion hat auch eine kameragestützte Bewegungsmelder-Funktion.

Diese Funktion kann bei Änderung einer bestimmten Anzahl von Pixeln im Kamerabild die lokale oder externe Speicherung und ggf. das Versenden von Video- und Bilddateien per E-Mail oder auch FTP bzw. die Ausführung anderer Aktionen (Ausführung von Scripten) auslösen. Dadurch wird unser Raspberry Pi zum intelligenten Bewegungsmelder mit Videofunktion und hochauflösenden Fotos.

Im nächsten Schritt des Tutorials besorgen wir uns das Software-Paket Motion, installieren es und richten es ein. Wir schiessen unser erstes Testfoto mit der Kamera und lernen die verschiedenen Einstellungen von Motion kennen.

Zum Schluss bauen wir die Kamera in ein passendes Gehäuse ein, damit unsere neue Video-, Überwachungs- und Webcam auch sicher verpackt ist und uns lange erhalten bleibt.

Vorbereitungen und Werkzeuge

Grundlage für dieses Projekt ist ein Raspberry Pi mit vollständig eingerichtetem Betriebssystem. Im Idealfall befindet sich der Pi schon per WLAN in Eurem Netzwerk und sowohl SSH als auch XRDP sind schon eingerichtet und funktionieren.

Wie Sie diese Vorbereitungen am besten erledigen, können Sie unter dem Hauptmenüpunkt "Grundlagen" in den einzelnen Tutorials nachlesen und nachvollziehen.

Für alle Einrichtungen und Arbeiten an der Software (egal ob Anwendungen oder Betriebssystem) empfiehlt es sich eine Netzwerkverbindung mit Kabel herzustellen. Die ist im Allgemeinen schneller und jedenfalls auf fallsicherer als eine WLAN-Verbindung. Grundsätzlich erfüllt natürlich auch eine solide WLAN-Verbindung alle Voraussetzungen, die für eine Einrichtung notwendig sind.

Obwohl das Tutorial sich grundsätzlich auf die 8MP (Megapixel) Kamera bezieht, funktioniert es grundsätzlich auch mit fast allen anderen Kameras, die für den Raspberry Pi im Handel oder bei uns im Shop erworben werden können.

Das Softwarepaket Motion funktioniert übrigens auch mit USB-Kameras und ist, wenn man sich weitergehend

damit beschäftigt, auch dazu in der Lage andere Motion-Cams zu steuern.

Spezielle Werkzeuge werden für die Montage der Kamera nicht benötigt. Der für die Verschraubung des Gehäuses benötigte Kreuzschlitzschraubendreher liegt dem Gehäuse schon bei. Die Kamera selbst wird im Kunststoffgehäuse mit einer Rasthalterung befestigt.

Kamera mit dem Raspberry Pi verbinden, aktivieren und testen

Bei der Verbindung zwischen Pi und Kamera kann man eigentlich gar nichts verkehrt machen. Wichtig ist, dass Ihr, bevor Ihr irgendwas am Pi steckt oder lötet oder anderweitig kabela wollt, den Pi ordentlich herunterfährt und stromlos macht. Sonst riskiert Ihr ernsthafte Beschädigungen an der Elektronik, die im schlimmsten Fall, den Raspberry Pi zerstören.

Jetzt nehmt Ihr einfach das mit der Kamera gelieferte Flachbandkabel und steckt es in den passenden Steckverbinder, der annähernd mittig auf der Raspberry Pi Platine angeordnet ist. Ein kurzer Klick und der Steckverbinder ist eingerastet und hält das Kabel, wenn es nicht zu stark belastet wird, sicher in seiner Position. Damit ist die mechanische Arbeit für diesen Schritt schon erledigt.

Jetzt schalten wir noch den Raspberry Pi ein und warten, bis er gestartet ist. Um den Kamera-Anschluss am Pi zu aktivieren, benötigen wir das Kommandozeilen-Programm Raspi-Config. Dafür gibt es zwar eine Entsprechung im Desktop, der Einfachheit halber, da es sowohl über eine SSH-Verbindung als auch im Desktop funktioniert, erkläre ich Euch die Einrichtung per Kommandozeilen-Tool raspi-config.

Wenn Ihr also direkt am Desktop des Pi oder per XRDP am Remote-Desktop arbeitet, müsst Ihr zunächst ein Terminal-Fenster öffnen.

Ab hier sind alle Schritte gleich.

Ihr öffnet das Programm raspi-config mit dem Kommando:

```
pi@rechnername: ~ $ sudo raspi-config (enter)
```

In dem nun angezeigten Menü wählt Ihr den Punkt "Interfacing Options" aus (enter).

Jetzt wählt den Menüpunkt "Camera" aus (enter).

Die Frage "Would you like the camera interface to be enabled?" (Möchten Sie das Kamera-Interface einschalten?) beantworten wir mit "Ja" (enter).

Danach wählen wir noch den Punkt "finish" (beenden) aus (enter) und schon ist die Kamera-Schnittstelle aktiviert.

Schliesslich booten wir den Raspberry Pi noch neu:

```
pi@rechnername: ~ $ sudo reboot (enter).
```

Wenn der Pi neu gestartet ist, verbinden wir uns ggf. neu und öffnen wieder ein Terminal-Fenster.

Jetzt können wir die neu installierte Kamera testen. Dazu bietet uns der Raspberry Pi wieder ein Kommandozeilen-Tool mit dem Namen "raspistill".

Das Kommando, um ein Foto mit der Kamera zu machen lautet:

```
pi@rechnername: ~ $ raspistill -o testbild.jpg (enter)
```

Jetzt sollte an der Kamera die rote LED kurz aufleuchten und wieder ausgehen. Wenn dann alles funktioniert hat, befindet sich jetzt im Verzeichnis /home/pi eine Bilddatei mit dem Namen "testbild.jpg".

Solltet Ihr eine solche Datei nicht finden oder hat vielleicht die LED gar nicht aufgeleuchtet? Fahrt bitte den Pi noch einmal komplett herunter und schaltet ihn aus. So viele Fehlerquellen gibt es nicht, also löst noch einmal alle Verbindungen zwischen Kamera und dem Raspberry Pi und verbindet sie neu. Achtet bitte darauf, dass sich die Kabel richtig herum in den Steckverbindern befinden, bevor Ihr sie arretiert.

Achtung!!! Auch der Stecker am Flexkabel der Kamera kann mal Kontaktprobleme machen. Also nicht vergessen, auch diese potenzielle Fehlerquelle zu überprüfen.

Danach noch einmal hochfahren und wieder ausprobieren. Wenn es dann immer noch nicht funktioniert, kontrolliert noch einmal die Einstellungen in raspi-config und korrigiert ggf. hier nach.

Jetzt sollten wir zum nächsten Schritt wechseln.

Softwarepaket Motion und RaspiCam als Videogerät installieren und einrichten

Das Werkzeug, mit dem wir jetzt aus unserer Raspi-Cam eine hochwertige Video-, Überwachungs- oder Webkamera machen, ist ein Software-Paket mit dem Namen "motion".

Zuerst einmal installieren wir dieses Softwarepaket auf unserem Raspberry Pi. Dazu verbinden wir uns wieder mit dem Pi und öffnen ein Terminal-Fenster. Mit der folgenden Befehlszeile laden wir das Paket herunter und installieren es:

```
pi@rechnername: ~ $ sudo apt-get install motion (enter)
```

Fragen, ob Ihr das Paket wirklich installieren möchtet, könnt Ihr ruhig mit J wie ja beantworten.

Das dauert dann ein kleines Weilchen. Zwischendurch werdet Ihr noch gefragt, auf welchem Kontinent und in welchem Land Ihr Euch befindet. Das beantwortet Ihr einfach wahrheitsgemäss und dann geht es weiter.

Nachdem jetzt also das Paket motion installiert ist, müssen wir noch dafür sorgen, dass unsere Kamera auch als Videogerät zur Verfügung steht.

Dafür müssen wir noch ein paar Zeilen in die Console tippen. Bei den Kommandos kommt ein paar mal die Buchstaben/Zahlenkombination "v4l2" vor. Dabei steht das v4l für Video for Linux. Das Zeichen "l" ist also ein kleines "L" und keine Ziffer oder auch kein grosses "l wie lgor". Und jetzt noch die Kommandos:

```
pi@rechnername : ~ $ sudo modprobe v4l2_common (enter)
pi@rechnername : ~ $ sudo modprobe bcm2835-v4l2 (enter)
```

Damit hätten wir zwar für jetzt ein Videogerät. Jetzt müssen wir noch dafür sorgen, dass uns dieses Gerät auch bei jedem Neustart zur Verfügung steht. Das machen wir so:

```
pi@rechnername : ~ $ echo "v4l2_common" | sudo tee -a /etc/modules (enter)
pi@rechnername : ~ $ echo "bcm2835-v4l2" | sudo tee -a /etc/modules (enter)
```

```
pi@rechnername : ~ $ sudo reboot (enter)
```

Und wir warten wieder, bis unser Raspberry Pi neu gebootet hat.

Nach dem Neustart schauen wir uns dann mal an, ob unsere Kamera auch als Videogerät zur Verfügung steht:

```
pi@rechnername : ~ $ ls /dev/video* (enter)
```

Hier müsste uns jetzt die Kamera als "/dev/video0" angezeigt werden. Das wäre toll. Jetzt sorgen wir noch dafür, dass motion bei jedem Systemstart als Dienst gestartet wird. Dazu bearbeiten wir noch kurz die Datei "/etc/default/motion":

```
pi@rechnername : ~ $ sudo nano /etc/default/motion (enter)
```

Hier ändern wir den Eintrag "start_motion_daemon" von "no" auf "yes".

Jetzt noch STRG+o (enter) zum Speichern und STRG+x zum Beenden und unser Motion-Paket sollte mit der Raspi-Cam funktionieren. Zuerst starten wir unseren Raspberry Pi neu:

```
pi@rechnername : ~ $ sudo reboot (enter)
```

Nach dem Neustart finden wir heraus, ob der Dienst motion funktioniert. Das geht so:

```
pi@rechnername : ~ $ sudo service motion (enter)
```

Nachdem jetzt unser Motion-Daemon funktioniert, können wir uns auch schon erste Ergebnisse unserer Arbeit ansehen. Motion bringt einen eigenen Webserver mit, über den wir nicht nur den Dienst konfigurieren können. Vielmehr bietet er die Möglichkeit, die Kamerabilder als Stream anzusehen.

Dazu starten wir im Desktop des Raspberry Pi den Webbrowser und rufen folgende Adresse auf:

```
http://localhost:8081
```

Auf dem Port 8081 läuft nämlich das Streaming der Kamerabilder von Motion. Für die Konfiguration des Paketes per Webserver ändert Ihr den Port einfach in 8080. Ohne weitere Konfiguration lässt sich dieser Stream jedoch zunächst nur aus dem Pi selbst ansehen.

Im nächsten Schritt schauen wir uns dann die Konfigurationsdatei von Motion etwas näher an.

Motion-Daemon einstellen und anpassen

Motion ist ein komplexer Dienst, der viele Funktionen und Möglichkeiten anbietet. Dem entsprechend gibt es natürlich auch umfangreiche Einstellungen, zwischen denen man wählen kann. Dafür benutzen wir die Konfigurationsdatei "/etc/motion/motion.conf".

```
pi@rechnername : ~ $ sudo nano /etc/motion/motion.conf (enter)
```

Da die Funktionen von Motion sehr umfangreich sind, werden wir uns in diesem Tutorial auf einige grundlegende Einstellungen beschränken und einige Parameter überspringen. Ihr müsst also durchaus, wenn Ihr den nächsten zu bearbeitenden Parameter sucht, ein bisschen scrollen.

Bevor wir jetzt an unserer motion.conf herumwerkeln noch ein Hinweis. Eine Änderung an der Datei wird erst dann wirksam, wenn wir den Dienst Motion nach Speicherung der motion.conf-Datei wieder neu starten. Das geht

so:

```
pi@rechnername: ~ $ sudo service motion restart (enter)
```

Deamon (Motion im Hintergrund als Dienst starten)

Damit motion als Dienst (Deamon) startet, müssen wir den Eintrag in der Abteilung "Deamon" von "off" auf "on" ändern:

```
#####  
# Daemon  
#####  
# Start in daemon (background) mode and release terminal (default: off)  
daemon on
```

Orientierung / Ausrichtung der Kamera - rotate

Der nächste Eintrag, um den wir uns kümmern ist der Parameter "rotate". Hier legen wir fest, ob und in welchem Winkel das Kamerabild gedreht werden soll. Wenn Ihr zum Beispiel eine Kamera in einem quer-Gehäuse längs (senkrecht) verbaut, ist natürlich Euer Bild gekippt. Das könnt Ihr korrigieren, indem Ihr (je nachdem in welche Richtung es gekippt ist) hier 90 oder 270 eintragt. Steht das Bild auf Kopf, dann müsst ihr halt 180 hier eintragen.

```
# Rotate image this number of degrees. The rotation affects all saved images as  
# well as movies. Valid values: 0 (default = no rotation), 90, 180 and 270.  
rotate 0
```

Auflösung (Breite und Höhe der Bilder und des Streams)

Breite und Höhe der Bilder werden über die Parameter "width" und "height" bestimmt und sollten im gleichen Verhältnis zueinander stehen, wie es die natürliche Auflösung der Kamera auch tut. Die acht Megapixel Kamera (Version 2) hat eine physikalische Auflösung von 3280 x 2464 Pixeln. Das entspricht einem Seitenverhältnis von fast genau 4:3. Die ältere fünf Megapixel Kamera, die natürlich auch funktioniert, hat eine physikalische Auflösung von 2592 x 1944 Pixeln. Auch hier kommen wir auf ein das gleiche Verhältnis. Somit könnt Ihr also problemlos alle Auflösungen, die über dieses Verhältnis verfügen benutzen. Je grösser Ihr das Format wählt, desto langsamer werden auch die Bildabfolgen. Ab 800 x 600 wird der Stream recht flüssig:

```
# Image width (pixels). Valid range: Camera dependent, default: 352  
width 1024
```

```
# Image height (pixels). Valid range: Camera dependent, default: 288  
height 768
```

Bewegungsmelderfunktion / Motion Detection

Die Bewegungsmeldung an der Kamera beruht darauf, dass die Änderung einer bestimmten Anzahl von Pixeln in zwei aufeinanderfolgenden Bildern als Bewegung gedeutet wird. Diese Anzahl kann man hier festlegen:

```
#####  
# Motion Detection Settings:  
#####
```

```
# Threshold for number of changed pixels in an image that  
# triggers motion detection (default: 1500)  
threshold 1500
```

Natürlich müsst Ihr die Anzahl dieser Pixel auf die von Euch gewählte Auflösung anpassen. Motion bietet zusätzlich die Möglichkeit, die Bewegungsmeldung (Motion-Detection) nur in bestimmten Bereichen des Kamerabildes einzusetzen. Dafür kann man eine Bildmaske hinterlegen, die diese Bereiche definiert. Doch das soll jetzt und hier nicht unser Thema sein.

Allgemeines zur Speicherung von Fotos bei Bewegungsmeldung

Das wir via Motion unserer Kamera am Webbrowser zuschauen können ist ja eine tolle Sache. Wenn wir unsere Kamera jedoch als Bewegungsmelder mit Kontroll- oder Alarmfunktion betreiben wollen, benötigen wir vielleicht Bilder als Beweismittel.

Hier versagen "normale" Kameras meistens wegen ihrer schlechten Auflösung. Wo also eine Full-HD-Kamera gerade einmal 1920x1080 Pixel aufzeichnet (das sind zwei Megapixel), kann unsere Raspi-Cam bis zu acht Megapixel Fotos machen.

Die hohe Qualität der Fotos kann, wenn man zum Beispiel Details oder nur kleine Ausschnitte aus Aufnahmen benötigt, entscheidend sein. Je grösser jedoch die Einzelbilder, desto weniger Bilder kann die Kamera in einer bestimmten Zeit machen. Und natürlich benötigen grössere Bilder auch mehr Platz auf unserer Speicherkarte. Wir müssen also entweder regelmässig aufräumen (alte Bilder löschen) oder wir brauchen nahezu grenzenlosen (externen) Speicherplatz in Form von Netzwerk-Laufwerken, Festplatten oder Cloudspeicher.

Ihr müsst also abwägen, ob Ihr einen flüssigen Stream mit niedrigerer Qualität wollt, oder ob Ihr eher alle paar Sekunden ein Bild mit hoher Qualität haben wollt. Das hängt ganz von dem Einsatzzweck Eurer Raspi-Cam ab. Probiert hier einfach aus, was für Euch richtig ist.

Sollen überhaupt Fotos bei Bewegung gespeichert werden?

Das und einige andere Optionen können wir mit dem Parameter "output_pictures" festlegen.

```
#####  
# Image File Output  
#####  
# Output 'normal' pictures when motion is detected (default: on)  
# Valid values: on, off, first, best, center  
# When set to 'first', only the first picture of an event is saved.  
# Picture with most motion of an event is saved when set to 'best'.  
# Picture with motion nearest center of picture is saved when set to 'center'.  
# Can be used as preview shot for the corresponding movie.  
output_pictures on
```

Qualität (Kompression) der gespeicherten Fotos

Je höher die Qualität, desto kleiner die Kompression. Auch hier heisst es wieder einen Kompromiss zwischen Geschwindigkeit und Qualität der Bilder zu finden. Je höher die Qualität, desto geringer die Kompression. Braucht Ihr also hochqualitative Bilder, dürft Ihr fast nicht oder gar nicht komprimieren und wählt am besten 80-100. Auch hier könnt Ihr ein bisschen probieren.

```
# The quality (in percent) to be used by the jpeg compression (default: 75)  
quality 75
```

Ausgabeverzeichnis für unsere Bilder und Videos (Basisordner)

Damit wir auch wissen, wo denn Motion unsere Bilder abspeichert, können wir hier auch das Basisverzeichnis (Basisordner) dafür festlegen. Für Bilder, Videos und Schnapsschüsse können dann noch diverse Unterordner erstellt werden. Dafür gibt es den Parameter "target_dir". Beachtet bitte, wenn Ihr hier etwas ändert, dass der

Dienst Motion dann auch Schreibrechte in dem von Euch eingetragenen Verzeichnis benötigt.

```
# Target base directory for pictures and films
# Recommended to use absolute path. (Default: current working directory)
target_dir /home/pi/motion
```

Ablageordner und Dateiname der durch Bewegung ausgelösten Bilder
Der Parameter "picture_filename" gibt und dann noch die Möglichkeit, die Dateinamen und den Unterordner für die so erzeugten Daten festzulegen. Zur Erklärung der verwendeten Kürzel soll die folgende Legende helfen:

```
#####
# Target Directories and filenames For Images And Films
# For the options snapshot_, picture_, movie_ and timelapse_filename
# you can use conversion specifiers
# %Y = year, %m = month, %d = date,
# %H = hour, %M = minute, %S = second,
# %v = event, %q = frame number, %t = camera id number,
# %D = changed pixels, %N = noise level,
# %i and %J = width and height of motion area,
# %K and %L = X and Y coordinates of motion center
# %C = value defined by text_event
# Quotation marks round string are allowed.
#####

# File path for motion triggered images (jpeg or ppm) relative to target_dir
# Default: %v-%Y%m%d%H%M%S-%q
# Default value is equivalent to legacy oldlayout option
# For Motion 3.0 compatible mode choose: %Y/%m/%d/%H/%M/%S-%q
# File extension .jpg or .ppm is automatically added so do not include this
# Set to 'preview' together with best-preview feature enables special naming
# convention for preview shots. See motion guide for details
picture_filename %v-%Y%m%d%H%M%S-%q
```

Stream auch für externe Rechner im Netz erlauben

Natürlich möchte man den Stream der Kamera auch auf anderen Rechnern im Netz ansehen. Damit unser Motion das auch anbieten müssen wir es zuerst erlauben. Dafür gibt es den Parameter "stream_localhost", der normalerweise eingeschaltet ist ("on") und damit den Stream ausschliesslich auf den Pi selbst, an dem die Kamera angeschlossen ist begrenzt. Einfach auf "off" setzen und schon ist Eure Cam im ganzen Netzwerk abrufbar. Das geht dann im Netz auf einem anderen Rechner über einen Webbrowser und dann die Adresse: <http://xxx.xxx.xxx.xxx:8081> wobei xxx.xxx.xxx.xxx natürlich für die IP-Adresse Eures Raspberry Pi steht und dementsprechend bei der Eingabe geändert werden muss.

```
# Restrict stream connections to localhost only (default: on)
stream_localhost off
```

Stream mit Passwortschutz sichern

Da aber eventuell nicht jeder im Netzwerk auch den Stream Eurer Motion-Cam ansehen können soll, ist es auch möglich, hier einen Passwortschutz einzubauen. Dafür gibt es einen Parameter für die Art des Passwort-Schutzes und einen für die Festlegung von Benutzernamen und Passwort. Setzt einfach den Parameter "stream_auth_method" auf "Basic authentication".
Set the authentication method (default: 0)
0 = disabled
1 = Basic authentication

```
# 2 = MD5 digest (the safer authentication)
stream_auth_method 0
```

Mit dem nächsten Parameter "stream_authentication" definieren wir dann noch, mit Doppelpunkt voneinander getrennt, den Benutzernamen und das Passwort für unseren Kamerastream.

```
# Authentication for the stream. Syntax username:password
# Default: not defined (Disabled)
; stream_authentication username:password
```

Wenn wir also Basic authentication als stream_auth_method und username:password als stream_authentication ausgewählt haben, können wir uns mit jedem Webbrowser eines Rechners in unserem Netz mit der folgenden Adresse in unserem Stream anmelden: `http://username:password@xxx.xxx.xxx.xxx:8081`

Damit soll es für jetzt auch erst einmal genug an Konfiguration für unsere Raspberry Pi Motion-Cam gewesen sein. Man kann auch Videos erstellen, Zeitrafferaufnahmen, man kann erstellte Fotos und Videos per E-Mail versenden oder per FTP ins Netz laden, es geht ganz ganz viel mit dieser Kamera zu machen. Auf die Upload-Funktionen komme ich bestimmt bei den Projekten PIR-Überwachungskamera und der Mikro-Alarmanlage, die ich in nächster Zeit noch als Tutorials mit einbauen werde zurück.

Einbau der Kamera ins Gehäuse

Nachdem wir nun unsere neue Raspberry-Pi-Motion-Überwachungs-Bewegungsmelder-Webcam eingerichtet haben, bauen wir das gute Stück jetzt auch noch in ein angemessenes Gehäuse ein.

Da mir hier eine Beschreibung per Text zu kompliziert und nicht anschaulich genug erscheint, zeige ich Euch die paar Schritte am besten in einem kurzen Video:

Kamera-LED ausschalten

Normalerweise und zum Testen ist es sicherlich gut, dass die Rasp-Cam über eine LED verfügt, die rot aufleuchtet, während die Kamera ein Foto oder Video aufnimmt.

Wenn die Kamera jedoch als Überwachungskamera oder zum Beispiel vor reflektierenden Oberflächen eingesetzt werden soll, ist diese Beleuchtung manchmal recht störend.

Aus diesem Grund ist es problemlos möglich, diese LED per Software abzuschalten. Ihr müsst einfach an die Datei: `/boot/config.txt` die folgende Zeile anhängen:

```
disable_camera_led=1
```

Backup / Datensicherung

Nachdem Ihr nun im Schweisse Eures Angesichts die Grundeinrichtung hinter Euch gebracht habt, möchte ich Euch zum Abschluss noch einen guten Rat geben.

Computersoftware, speziell Betriebssysteme sind ein extrem komplexes System an Komponenten, die im idealfall

auch noch perfekt zueinander passen.

Da wir aber im realen Leben existieren, ist der Idealfall oft nicht der, mit dem wir es zu tun bekommen. Darum hier mein gut gemeinter Rat:

Der Win32 Diskimager bietet euch neben der Funktion, Images auf eine SD-Karte zu schreiben, auch die Funktion, Images von einer SD-Karte zu lesen und zu speichern.

Da es ja meistens nicht mit der Grundinstallation getan ist, man also, ggf. auch experimentell, weiter installieren und konfigurieren muss, kann hier schnell mal etwas schiefgehen.

Wenn Ihr also einen Installationsschritt abgeschlossen habt und ihr einen Softwarestand der SD-Karte erreicht habt, der sicherungswert ist, weil er mal wieder viel Zeit gekostet hat und doch nur ein Zwischenschritt ist oder wenn Ihr direkt mit einer Installation fertig seid und alles perfekt funktioniert:

MACHT EUCH EIN IMAGE MIT DEM JEWEILIGEN SOFTWARESTAND

Es dauert im Durchschnitt nur ein paar Minuten, eine SD-Karte wieder zurückzusichern. Eine Neueinrichtung kann wieder Stunden benötigen. Speichert Euch die Imagedateien und benennt sie mit dem Datum und beschreibt den Softwarestand im Dateinamen, dann findet Ihr sie auch wieder, wenn Ihr sie sucht.

Ausserdem lässt sich eine bestehende Grundeinrichtung natürlich auch als Basis für verschiedene Pi oder verschiedene Konfigurationen Eures einen Pi nutzen.

Ich bedanke mich für Eure Aufmerksamkeit und wünsche Euch viel Spass beim Raspeln!