

# **Anleitung für die Konfiguration eines Kamera-Livestreams sowie Motion Detection für den Nistkasten**

Dokumenten ID: EBIRD-UM-010

Version: 0-16 vom 14. April 2017

Autor: Jan Grosser

[email@jan-grosser.de](mailto:email@jan-grosser.de)

<https://github.com/rzbrk/ebird>

## Inhaltsverzeichnis

1Einführung.....	3
2Voraussetzungen.....	3
3Software.....	4
3.1Konfiguration des Betriebssystems.....	4
3.1.1V4L-Treiber für die Kamera.....	4
3.1.2Benutzergruppe ebird.....	4
3.1.3Ausgabeverzeichnis für Bilder und Videos von motion.....	4
3.2motion.....	5
3.2.1Installation von motion.....	5
3.2.2Konfiguration von motion.....	5
3.2.3Starten und Testen von motion.....	6
Anlage 1: motion Parameter.....	7

# 1 Einführung

Diese Anleitung beschreibt die Einrichtung eines Kamera-Livestreams sowie einer Motion Detection (Bewegungserkennung) für den Raspberry Pi. Über den Livestream kann permanent das Bild der Kamera über eine Web-Schnittstelle angesehen werden. Die Bewegungserkennung speichert Bilder und Videos von der Kamera, sobald eine Bewegung detektiert wird.

Diese Anleitung basiert auf der Anleitung von [www.datenreise.de](http://www.datenreise.de).

Für die Erzeugung des Livestreams sowie der Bewegungserkennung wird die Software **motion** (<https://motion-project.github.io/>) verwendet und so konfiguriert, daß die Bilder und Videos in den Ordner **/var/ebird/cam** abgelegt werden. Die Bilder und Videos werden dort nach Tagen in Ordnern nach dem Format **YYY-MM-DD/** abgelegt. Für jedes von motion detektierte Event (registrierte Bewegung) wird ein Unterordner nach dem Format **event-YYY-MM-DD-HHMMSS-VV/** angelegt. VV ist dabei eine fortlaufende Event-Nummer. Auch ohne Bewegung wird alle 5 Minuten ein Snapshot erzeugt. Aus allen Snapshots wird tageweise ein Timelapse-Video erzeugt (**YYY-MM-DD/YYYYMMDD-timelapse.mpg**). Der jeweils aktuelle Snapshot steht unter **/var/ebird/cam/lastsnap.jpg** zur Verfügung.

# 2 Voraussetzungen

Vorausgesetzt wird ein Raspberry Pi mit installiertem Raspbian (<https://www.raspberrypi.org/downloads/raspbian/>) oder vergleichbarem Betriebssystem. Zum Installieren der erforderlichen Software sollte eine Internetverbindung bestehen. Loggen Sie sich via SSH auf dem Raspberry Pi ein.

An dem Raspberry Pi sollte eine Raspi Kamera angeschlossen sein. Dies erfolgt über das CSI-Flachbandkabel der Kamera mit dem CSI-Stecker auf dem Raspberry Pi. Die Kamera nicht an- oder abstecken, falls der Raspberry Pi läuft!

Die Raspi Kamera muß nun im Konfigurations-Programm **raspi-config** aktiviert werden. Starten Sie hierzu das Konfigurations-Programm über die Kommandozeile wie folgt:

```
pi $ sudo raspi-config
```

Aktivieren Sie nun unter Interfacing Options → Camera die Kamera.

Nun kann die Kamera getestet werden. Nehmen Sie bspw. Ein Einzelbild auf:

```
pi $ raspistill -o test.jpg
```

Die Datei sollte ohne Fehlermeldung erzeugt werden.

## 3 Software

### 3.1 Konfiguration des Betriebssystems

#### 3.1.1 V4L-Treiber für die Kamera

Für **motion** muß die Kamera mit dem Video4Linux-Treibern funktionieren und das Device **/dev/video0** angelegt sein. Hierzu müssen die zwei Treiber **v4l2\_common** und **bcm2835-v4l2** in der Datei **/etc/modules** eingetragen sein. Das stellen wir über folgende Befehle sicher:

```
pi $ cat /etc/modules | grep -e „^v4l2_common$“ \  
    || sudo cat „v4l2_common“ >> /etc/modules  
pi $ cat /etc/modules | grep -e „^bcm2835-v4l2$“ \  
    || sudo cat „bcm2835-v4l2“ >> /etc/modules
```

Wenn der Befehl

```
pi $ sudo lsmod
```

nicht die beiden Treiber anzeigt, sind sie aktuell nicht geladen. Geladen werden können diese mit dem Befehl:

```
pi $ sudo modprobe v4l2_common && sudo modprobe bcm2835-v4l2
```

Wenn die Treiber ordnungsgemäß geladen sind, wird das Device **/dev/video0** angelegt.

#### 3.1.2 Benutzergruppe ebird

Nun prüfen wir, ob die Benutzergruppe **ebird** vorhanden ist. Falls nicht, erzeugen wir diese:

```
pi $ cat /etc/group | grep ebird
```

Falls die Gruppe nicht angezeigt werden sollte, legen wir sie an:

```
pi $ sudo groupadd ebird
```

### 3.1.3 Ausgabeverzeichnis für Bilder und Videos von motion

Nun legen wir noch ein Verzeichnis für die Kamera-Bilder und -Videos an und richten Schreibberechtigung für alle Benutzer der Gruppe **ebird** ein:

```
pi $ sudo mkdir -p /var/ebird/cam
pi $ sudo chgrp ebird /var/ebird/cam
pi $ sudo chmod g+w /var/ebird/cam
```

## 3.2 motion

### 3.2.1 Installation von motion

Die Software **motion** kann über die apt-Paketverwaltung installiert werden:

```
pi $ sudo apt-get update
pi $ sudo apt-get install motion
```

Mit der Installation von motion sollte auch ein gleichnamiger Benutzer **motion** angelegt worden sein, was wir mit folgendem Kommando überprüfen können:

```
pi $ cat /etc/passwd | grep motion
```

Nun fügen wir den Benutzer **motion** der Gruppe **ebird** zu, damit **motion** Bilder und Video in das oben erzeugte Verzeichnis **/var/ebird/cam** ablegen kann.

```
pi $ sudo gpasswd -a motion ebird
```

### 3.2.2 Konfiguration von motion

Motion soll als Daemon automatisch beim Systemstart aktiviert werden. Hierzu ist in der Datei **/etc/default/motion** die Option **start\_motion\_daemon** auf den Wert **yes** zu setzen. Die Datei kann dazu mit dem Editor **nano** bearbeitet werden:

```
pi $ sudo nano /etc/default/motion
```

Das Verhalten von motion wird über die Konfigurationsdatei **/etc/motion/motion.conf** gesteuert. Sie enthält eine Reihe von Parametern. Die Anpassung der Konfigurationsdatei für dieses Nistkasten-Projekt findet sich im Software Repository auf Github unter dem Link:

[https://github.com/rzbrk/ebird/blob/master/10\\_cam/config/motion.conf](https://github.com/rzbrk/ebird/blob/master/10_cam/config/motion.conf). Sie kann wie folgt auf den Raspberry Pi übertragen werden:

```
pi $ sudo cp /etc/motion/motion.conf \
    /etc/motion/motion.conf.old # Alte Config sichern

pi $ sudo wget \
    https://github.com/rzbrk/ebird/blob/master/10_cam/config/motion.conf \
    -o /etc/motion/motion.conf
```

Einige Parameter sind in Anlage 1 erläutert.

### 3.2.3 Starten und Testen von motion

Motion kann nun über folgenden Befehl gestartet werden:

```
pi $ sudo systemctl restart motion
```

Es empfiehlt sich, in einem weiteren Terminal bzw. einem weiteren Pane eines terminal-Multiplexers (z.B. **tmux** oder **screen**) das Programm **journalctl** (Befehl: **sudo journalctl -f**, beenden mit Ctrl+c) laufen zu lassen, um die Meldungen des Systems und des Daemons motion zu verfolgen. Dort erhält man ggf. wertvolle Hinweise zu Fehlern.

Der Stream kann nun über einen Arbeitsplatz-Computer im gleichen Netzwerk über die Adresse **http://<hostname>:8081** abgerufen werden. **<hostname>** ist hierbei durch den Hostnamen oder die IP-Adresse des Raspberry Pi zu ersetzen.

## Anlage 1: motion Parameter

Parameter aus der Datei `/etc/motion/motion.conf`. Siehe [Link github.com](#).

Parameter	Wert	Erläuterung
width	640	Image width (pixels)
height	480	Image width (pixels)
framerate	10	Maximum number of frames to be captured per second.
max_movie_time	0	Video length infinite
ffmpeg_timelapse	1	Creation of time lapse video enabled
snapshot_interval	300	Make automated snapshot every N seconds
text_event	%Y-%m-%d-%H%M%S	This option defines the value of the special event conversion specifier %C. The idea is that %C can be used filenames and text_left/right for creating a unique identifier for each event.
target_dir	/var/ebird/cam	Target base directory for pictures and films
snapshot_filename	%Y-%m-%d/snapshots/%Y%m%d%H%M%S-snapshot	File path for snapshots relative to target_dir
picture_filename	%Y-%m-%d/event-%C-%v/%v-%Y%m%d%H%M%S-%q	File path for motion triggered images relative to target_dir
movie_filename	%Y-%m-%d/event-%C-%v/%v-%Y%m%d%H%M%S	File path for motion triggered ffmpeg films relative to target_dir
timelapse_filename	%Y-%m-%d/%Y%m%d-timelapse	File path for timelapse movies relative to target_dir
stream_port	8081	The mini-http server listens to this port for requests http://<hostname>:8081
stream_motion	on	Output frames at 1 fps when no motion is detected
stream_maxrate	1	Maximum framerate for stream streams
stream_localhost	off	Restrict stream connections to localhost only

