

Anleitung für die Konfiguration einer Scrubbing-Routine

Dokumenten ID: EBIRD-UM-020

Version: 0-13 vom 14. April 2017

Autor: Jan Grosser

email@jan-grosser.de

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

Inhaltsverzeichnis

1Einführung.....	3
2Voraussetzungen.....	3
3Software.....	3
3.1Klonen des ebird Repositories von github.....	3
3.2Kopieren der Dateien ins System.....	3
3.3Konfiguration des Schedulers cron.....	4
Anlage 1.....	5
Anlage 2.....	6
3.4Freien Speicherplatz anzeigen.....	6
3.5Größe von Verzeichnissen und Dateien in einem Verzeichnis ausgeben lassen.....	6
3.6Anzahl der Dateien in einem Verzeichnis anzeigen lassen.....	6
3.7Jüngstes Video (AVI Datei) anzeigen lassen.....	7

1 Einführung

Diese Anleitung beschreibt die Einrichtung einer sog. Scrubbing-Routine für den Raspberry Pi. Voraussetzung ist die erfolgreiche Konfiguration der Software **motion** (siehe EBIRD-UM-010). Die Software **motion** erzeugt bei Detektion von Bewegungen in einem Kamerabild Bilder und Videos. Diese Daten werden im Verzeichnis **/var/ebird/cam** abgelegt.

Wird motion über eine gewisse Zeit betrieben, wachsen die Daten in dem Verzeichnis **/var/ebird/cam** stetig an. Die Scrubbing-Routine soll verhindern, daß der Speicher des Raspberry Pis vollgeschrieben wird. Sinkt der freie Speicherplatz auf dem Speicher unter einem kritischen Wert, löscht die Scrubbing-Routine die ältesten Dateien in **/var/ebird/cam**, bis wieder ausreichend Speicherplatz zur Verfügung steht.

Die Scrubbing-Routine wird regelmäßig als cron-Job ausgeführt.

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. Außerdem wird ein konfigurierte Software motion vorausgesetzt, welche Bilder und Videos in das Verzeichnis **/var/ebird/cam** ausgibt. Loggen Sie sich via SSH auf dem Raspberry Pi ein.

3 Software

3.1 Klonen des ebird Repositories von github

Grundsätzlich ist es egal, wohin Sie das ebird-Repository klonen. Es empfiehlt sich allerdings, das Repository zentral im System abzulegen. Gehen Sie hierzu gemäß der Anleitung in Anlage 1 vor. Das Repository wird nun wie folgt in /usr/src geklont:

```
pi $ cd /usr/src
```

```
pi $ git clone https://github.com/rzbrk/ebird.git
```

3.2 Kopieren der Dateien ins System

Einige Dateien (Skript und Konfigurationsdatei) müssen nun ins System kopiert werden:

```
pi $ sudo cp /usr/src/ebird/20_scrubbing/scripts/scrubbing \
    /usr/local/bin/.

Pi $ sudo chmod +x /usr/local/bin/scrubbing

pi $ sudo mkdir -p /etc/ebird

pi $ sudo cp \
    /usr/src/ebird/20_scrubbing/config/scrubbing.conf \
    /etc/ebird/scrubbing.conf
```

3.3 Konfiguration des Schedulers cron

Das Scrubbing-Script soll regelmäßig, alle 4 Stunden ausgeführt werden. Hierzu wird ein Job im Scheduler cron konfiguriert:

```
pi $ echo "0 */4 * * * motion nice -10 \
    /usr/local/bin/scrubbing" >> /etc/crontab
```

Wichtige Hinweise:

- Die Scrubbing-Routine ist potentiell gefährlich! Sie sollte mit möglichst niedrigen Rechten gestartet werden, also bspw. nie als **root**! Im obigen Beispiel wird die Routine mit dem User **motion** gestartet.
- Der Aufruf der Routine erfolgt mit **nice -10** gestartet. Dadurch läuft das Skript mit niedriger Priorität und bremst andere laufende Prozesse möglichst wenig aus.
- Die Routine wird wegen **0 */4 * * *** alle 4 Stunden ausgeführt.

Jetzt kann man den Scheduler **cron** neu starten:

```
pi $ sudo systemctl restart cron
```

Einige nützliche Kommandos zur Beobachtung der Dateien im Ordner `/var/ebird/cam` sind in Anlage 2 aufgeführt.

Anlage 1

Diese Anleitung beschreibt die Einrichtung des Ordners **/usr/src** als zentralen Ordner für das Klonen von Repositories.

```
pi $ cat /etc/group | grep -e "^dev:" || sudo groupadd dev  
pi $ sudo gpasswd -a pi dev
```

Es können mit obigen Befehl auch weitere Benutzer der Gruppe **dev** zugeordnet werden. Ersetzen Sie hierzu **pi** mit dem Namen des anderen Benutzers.

```
pi $ sudo mkdir -p /usr/src  
pi $ sudo chgrp dev /usr/src  
pi $ sudo chmod g+w /usr/src
```

Nun kann noch ein symbolischer Link im Home-Verzeichnis eines Benutzers auf dieses Verzeichnis angelegt werden:

```
pi $ ln -s /usr/src ~/src
```

Anlage 2

Dieses Kapitel listet einige nützliche Kommandos zur Kontrolle des Bilder- und Video-Verzeichnisses von **motion** auf. Sie können aber (abgewandelt) auch für andere Verzeichnisse bzw. Aufgaben verwendet werden.

3.4 Freien Speicherplatz anzeigen

```
pi $ df -h
```

Dateisystem	Größe	Benutzt	Verf.	Verw%	Eingehängt auf
/dev/root	15G	1,2G	13G	9%	/
devtmpfs	182M	0	182M	0%	/dev
tmpfs	186M	0	186M	0%	/dev/shm
tmpfs	186M	4,4M	182M	3%	/run
tmpfs	5,0M	4,0K	5,0M	1%	/run/lock
tmpfs	186M	0	186M	0%	/sys/fs/cgroup
/dev/mmcblk0p1	63M	21M	42M	33%	/boot

3.5 Größe von Verzeichnissen und Dateien in einem Verzeichnis ausgeben lassen

```
pi $ du -hs /var/ebird/cam
```

```
27M    /var/ebird/cam
```

3.6 Anzahl der Dateien in einem Verzeichnis anzeigen lassen

```
pi $ find /var/ebird/cam -type f | wc -l
```

```
373
```

3.7 Jüngstes Video (AVI Datei) anzeigen lassen

```
pi $ find /var/ebird/cam -type f -name "*.avi" -print0 | \
    xargs -0 ls -lt | head -1
```

Ersetzt man im `ls`-Befehl die Option `-lt` durch `-ltr`, so wird das älteste Video angezeigt.