

# OctoPi verzögert abschalten mit Tasmota



- mindestens Tasmota 9.2.0 nutzen

Um den 3D Drucker (und [OctoPi](#)) automatisch abzuschalten beim Shutdown kann man unter „Einstellungen → Server → System herunterfahren“ ein Script angeben das zum Shutdown ausgeführt wird.

Wir nutzten dies um eine [Tasmota](#)-gefäschte WLAN Steckdose verzögert abschalten zu lassen.



Der Trick besteht darin, dass auf dem Tasmota Device eine RULE erstellt wird, die Zeitverzögert die Steckdose ausschaltet.

Dieses Script richtet die RULE automatisch als RULE2 ein und löst im Anschluss den Trigger aus.



dieses Script **überschreibt** eine bestehende **RULE** (Regel).

## Script

Das Script legen wir in **/home/pi** ab und rufen es dann so auf (Angabe unter genannter Einstellung):

```
sudo /home/pi/shutdown_steckdose_off.sh 180
```

## OctoPrint Einstellungen

### DRUCKER

[Serielle Verbindung](#)

[Druckerprofile](#)

[Temperaturen](#)

[Terminalfilter](#)

[GCODE Scripts](#)

### FUNKTIONEN

[Funktionen](#)

[Webcam & Zeitraffer](#)

[Zugangsbeschränkung](#)

[GCode Viewer](#)

[API](#)

[Application Keys](#)

### OCTOPRINT

[Server](#)

Erlaube, das Webinterface in einen Frame oder IFrame einbinden

[Needs restart](#)

## Befehle

OctoPrint neustarten

[sudo service octoprint restart](#)

System neustarten

[sudo shutdown -r now](#)

System herunterfahren

[sudo /home/pi/shutdown\\_steckdose\\_off.sh 180](#)

## Onlineprüfung

Wenn die Onlineprüfung aktiviert ist, wird OctoPrint in regelmäßigen Abständen prüfen, ob der Internet verbunden ist. Das ist nützlich um resourcenintensive Operationen zu unterbinden (z.B. Updates), wenn es bereits klar ist, dass diese Operationen nicht durchgeführt werden können.

- Der RPI wird kontrolliert beendet und nach **180** Sekunden schaltet die Steckdose aus.

Natürlich muss das Script ausführbar sein:



Linux Konsole:

```
chmod a+x /home/pi/shutdown_steckdose_off.sh
```

```
#!/bin/sh
#set -x #debug aktivieren
# Script um beim Shutdown von OctoPi auch den Drucker (incl. Octopi)
abzuschalten
# Da ein RaspberryPi sich ja nicht selbst ausschalten kann macht das hier
# eine Steckdose mit Tasmota
# Benutzung:
# /home/pi/shutdown_steckdose_off.sh 90
# Schaltet die Steckdose mit 90 Sekunden verzögerung ab.

# IP der Tasmota Steckdose
ip=192.168.178.210

off=$1

#Verwendeter 1. RuleTimer
timer=1
#Verwendete 2. RULE (Regel)
rulenum=2

#Rule Power OFF erste Steckdose falls es mehrer gibt.
```

```

rule="ON Rules#Timer=$timer DO Power1 off ENDON"

if [ -n "$off" ]
then
  # Erstmal prüfen ob die notwendige Rule schon existiert und OK ist
  result=`wget -q -O - http://$ip/cm?cmnd=rule$rulenumber | sed
's/.*"Rules"://` | sed 's/".*//'
  # Wenn nicht passt dann wird die rule neu erstellt/überschrieben.
  if [ "$rule" != "$result" ]
  then
    setrule=`echo $rule | sed 's/ /%20/g' | sed 's/#/%23/g'`
    result=`wget -q -O - http://$ip/cm?cmnd=rule$rulenumber%20$setrule` 
    result=`wget -q -O - http://$ip/cm?cmnd=rule$rulenumber%20N` 
  fi
  if [ $off -gt 0 ]
  then
    #Timer setzen - nach Ablauf der Zeit schaltet die Steckdose aus.
    wget -q -O /dev/null http://$ip/cm?cmnd=Backlog%20rule$timer$timer%20$off
  else
    #Direkt ausschalten - weniger tolle Idee...
    wget -q -O /dev/null http://$ip/cm?cmnd=Backlog%20Power1%20off
  fi
fi

echo "OctoPi Shutdown"
sleep 1
/sbin/shutdown -h +1
exit 0

```

## sudoers

das Script muss noch in sudoers eingetragen werden, damit es ohne eingabe eines Passwortes funktioniert

```

pi@octopi:~ $ sudo editor /etc/sudoers.d/010_pi-nopasswd
pi ALL=(ALL) NOPASSWD: /home/pi/shutdown_steckdose_off.sh

```



unter vi oder vim (apt install vim) kann das schreiben mit :w! erzwingen werden

From:

<https://www.myworkroom.de/> - **Sodele**



Permanent link:

[https://www.myworkroom.de/p-rr:octopi\\_verzoegert\\_abschalten](https://www.myworkroom.de/p-rr:octopi_verzoegert_abschalten)

Last update: **2021/01/26 08:16**