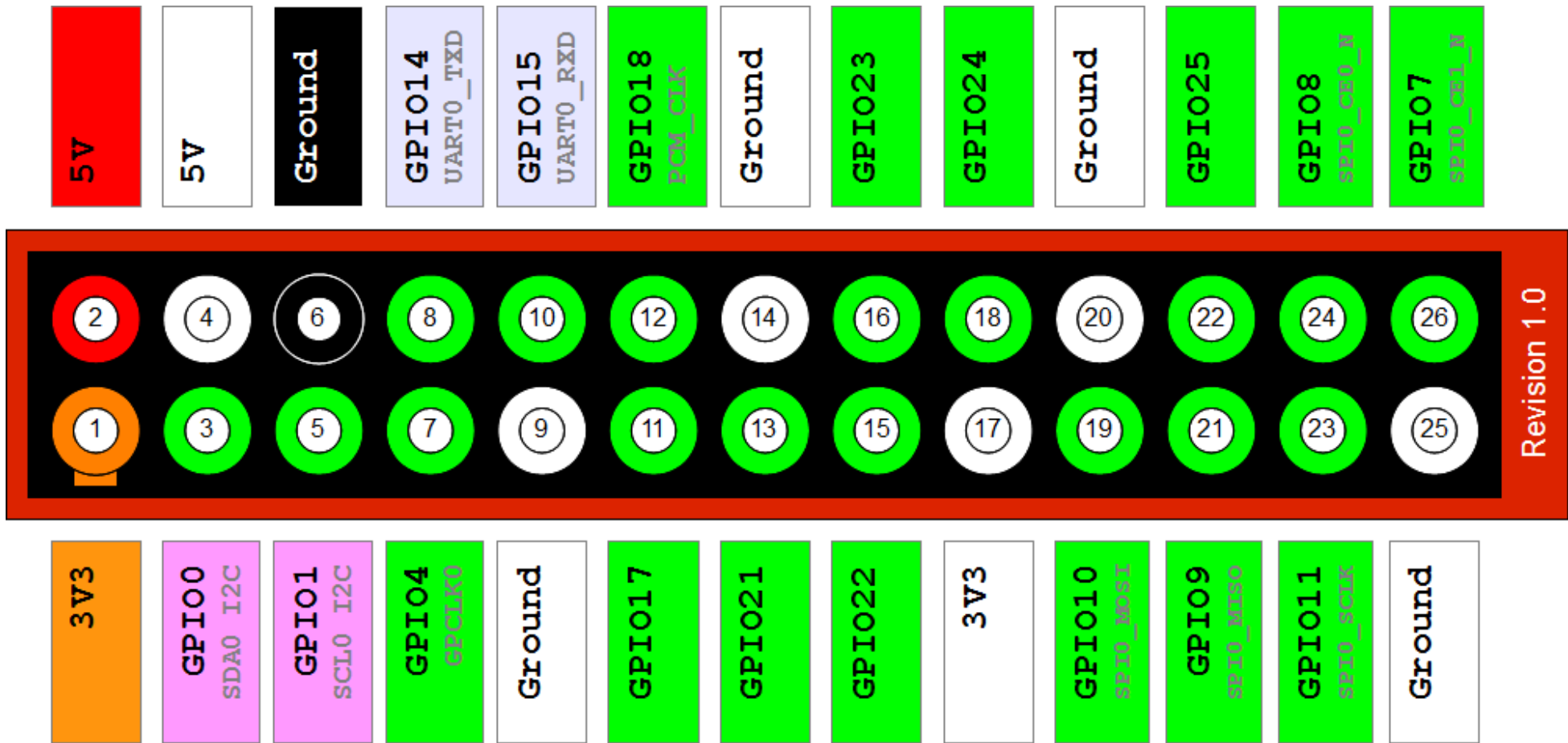


Funksteckdosen via Raspberry Pi ansteuern

Was wird benötigt?

- Raspberry Pi mit SD-Karte und Betriebssystem (z.B. Raspbian)
- 433MHz Sender (ca. 2 € bei dx.com)
- ein paar Jumper-Kabel (ca. 3 € bei dx.com)
- Funksteckdosen z.B. vom Typ Elro AB440S

GPIO-Anschlüsse des Raspberry Pi (Mod. B, Rev. 1)



(Bitte je nach Bauart des Raspberry überprüfen, wie die GPIO-Belegung ist.)

Transmitter an den Raspberry anschließen



ATAD (Data) → GPIO17 (Pin 11)

VCC → 5 V (Pin 2)

GND → Ground (Pin 6)

ANT → optionale Antenne

Jetzt geht's an die Software

- per SSH auf den Raspberry einloggen
ssh pi@IP-Adresse; Passwort: raspberry
- sudo apt-get update
- sudo apt-get install git-core g++ make
- git clone git://git.drogon.net/wiringPi
(installiert das Paket wiringPi zur Ansteuerung des Senders)
- cd wiringPi
- ./build

- `cd ~`
- `git clone git://github.com/xkonni/raspberry-remote.git`
(installiert das Paket `raspberry-remote` für die Ansteuerung über die Web-Oberfläche)
- `cd raspberry-remote`
- `make send`
(bei `raspbmc/osmc` sind ggf. noch Pakete wie `g++` zum Kompilieren nachzuinstallieren)

Bereits jetzt ist technisch gesehen die Ansteuerung der Steckdosen über die Konsole möglich.

ABER: Erstmal den Hauscode der Steckdosen festlegen

Die Pins 1 – 5 stellen den Hauscode dar. Die Pins A – D die Steckdosenummer.

Hauscode:
oben=1
unten=0

Steckdosenummer:
Bitte aufpassen, dass hier
nur ein Pin oben ist!



(Beispielfoto – Hauscode 10110 Steckdose 1)

Mit dem Kommando

```
sudo ./send 01100 3 1
```

kann man jetzt die Steckdose anschalten

01100 steht dabei für den Hauscode, 3 für die Steckdosenummer und 1 für an.

Das war ja schon ganz nett, aber jetzt geht's ans...

Webinterface:

- `sudo apt-get install apache2 php5`
- `sudo mkdir /var/www/remote`
- `sudo mv ~/raspberrry-remote/webinterface/* /var/www/remote`
(Inhalt von Ordner ~/raspberrry-remote/webinterface nach /var/www/remote verschoben)
- `cd /var/www/remote`
- `nano config.php`
(IP-Adresse des Raspberry Pi und die Steckdosen anpassen)
- `cd /home/pi/raspberrry-remote/`

- nano daemon.cpp
Hier nach dem Begriff „nPlugs“ suchen und den Wert auf „1110“ abändern, um den Bereich der Hauscodes zu erweitern.
- sudo make daemon

Wenn man jetzt mit *sudo ./daemon &* den Daemon startet, sollte über das Web-Interface bereits die Steuerung möglich sein.

Aber vorher legen wir noch fest, dass der Daemon automatisch startet bzw. sich neu startet, falls er mal abstürzt.

- sudo mv daemon daemon_remote
- cd /home/pi/

- `sudo touch /home/pi/daemon_restart.sh`
- `sudo nano /home/pi/daemon_restart.sh`

folgendes eintragen:

```
#!/bin/bash  
# Check if daemon_remote running  
if ! ps aux | pgrep "daemon_remote" > /dev/null; then  
    cd /home/pi/raspberry-remote/  
    ./daemon_remote &  
    exit 0  
fi
```

- `sudo chmod +x daemon_restart.sh`
- `sudo crontab -e`

folgendes eintragen

```
@reboot cd /home/pi/ && ./daemon_restart.sh  
*/15 * * * * cd /home/pi/ && ./daemon_restart.sh
```

- `sudo reboot`
(Raspberry Pi neu starten)
- In einem Browser `http://“IP-Adresse-des-Raspi“/remote` eingeben
und die Steckdosen steuern

Quelle: <http://alexbloggt.com/funksteckdosen-raspberry-pi-teil1/>