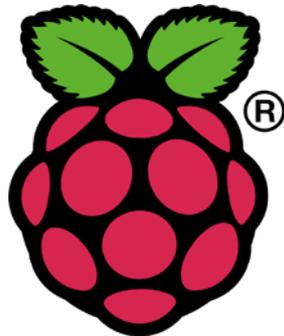


Projektbeschreibung Switcher

Beschreibung des Raspberry-Projekts „Schalten Funksteckdosen“ mit dem Raspberry Pi.



Inhaltsverzeichnis

1.	Raspberry Pi Projekt	1
2.	Abstract	1
3.	Anforderungen.....	1
4.	Fertige Lösung	2
5.	Hardware Aufbau.....	3
5.1	Pin Belegung Raspberry Pi	4
5.2	Verbindung zum Handsender.....	5
5.3	UPiS.....	5
5.4	Nobody Home Switch.....	6
6.	Software.....	6
6.1	Switcher-Programm.....	6
6.2	Control File.....	7
6.3	Switcher Client.....	7
6.4	Schaltzeiten anzeigen	8
7.	Daemonisierung	8
7.1	Switcher mit Daemon.....	8
7.2	Switcher ohne Daemon	9
8.	Interprocess Kommunikation.....	9
9.	Shutdown Pi	9
10.	Web-Interface	10
11.	Software Komponenten	10
12.	Links.....	11
13.	Anhang.....	12

1. Raspberry Pi Projekt

Der Raspberry Pi (genannt Pi) ist ein kreditkarten-kleiner und preisgünstiger (45 CHF) Linux Computer der sich seit seiner Einführung vor bald 4 Jahren grosser Beliebtheit erfreut und der bereits mehrere Millionen Mal verkauft wurde. Im Internet finden sich unzählige Projektbeschreibungen; alles Mögliche und Unmögliche wird mit diesem Micro-Computer gebaut. Der Initiator Eben Upton ist Engländer und in folgendem Video auf YouTube erklärt die Geschichte des Pi.

[Eben Upton on YouTube](#)

Der Pi hat 4 USB-Anschlüsse, einen Lan-Anschluss, ein HDMI-Anschluss für Monitor und einen Audio-Video-Ausgang. Ebenfalls auf dem Board ist ein Anschluss für die Pi-Camera - eine HD-fähige Kleinst-Camera für Fotos und Video. Was den Pi auszeichnet, ist die 40-polige Steckerleiste. Viele dieser Anschlüsse sind General Purpose Input/Output Pins, welche in Programmen angesteuert/gelesen werden können. Es gibt in der Zwischenzeit vielfältige Expansion Boards von unzähligen Anbietern. Der Pi konsumiert bloss 1 Watt Leistung: 5V und 200mA Strom.

Dieses Maschinchchen hat mich von Anfang an fasziniert, ich hatte mir in 2012 eines angeschafft. Nach einigem Probieren legte ich es auf die Seite - fand einfach kein wirkliches Projekt. Dies hat sich im Frühjahr 2014 geändert und es kam die Idee auf für eine Lampensteuerung bei Ferienabwesenheit.



Der Raspberry Pi Model B+

2. Abstract

This documentation describes a Raspberry Pi gadget that is used to switch lights or other appliances on/off. Up to 4 remote switches are supported, multiple on/off sequences for every switch per day are possible. Weekdays and on-off-times coded in XML file, defined sequence repeats ever week. XML file is parsed at reboot. With ‚At Home‘ switch to suppress switching when present. With webinterface to show current switch status.

3. Anforderungen

Bei Ferienabwesenheit sollen 4 Funksteckdosen (Typ ELRO) gemäss einem vorgegebenen Schema ein- und ausgeschaltet werden. Die Funksteckdosen werden in verschiedenen Räumen platziert und die geschalteten Lampen sollen Anwesenheit vortäuschen. Pro Dose soll jeder Wochentag (von Sonntag bis Samstag) ein eigenes Schaltprogramm haben und die Schaltprogramme werden jede Woche wiederholt. Die Schaltaktionen sollen extern des Programms in einem XML-File definiert sein. Der Raspberry soll mit einem Uninterruptible Power Supply ausgestattet sein.

Das Schaltprogramm (Python) soll als Linux-Daemon laufen, der bei Boot des Pi gestartet wird. Via Inter-process-Kommunikation kann ein Client-Programm den Schaltstatus abfragen.

Zudem soll via ein Web-Interface der aktuelle Schaltstatus abgefragt werden können.

4. Fertige Lösung

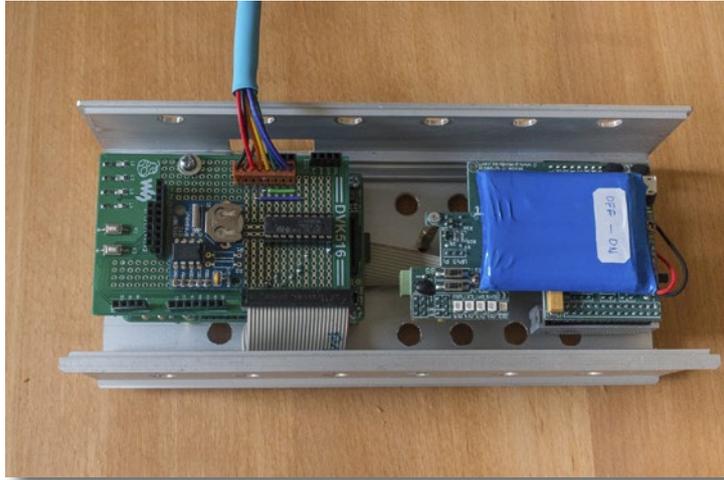
Der Aufbau der fertigen Lösung sieht so aus - ohne Gehäusedeckel



Switcher Hardware Aufbau



Die 4 schaltbaren Funksteckdosen

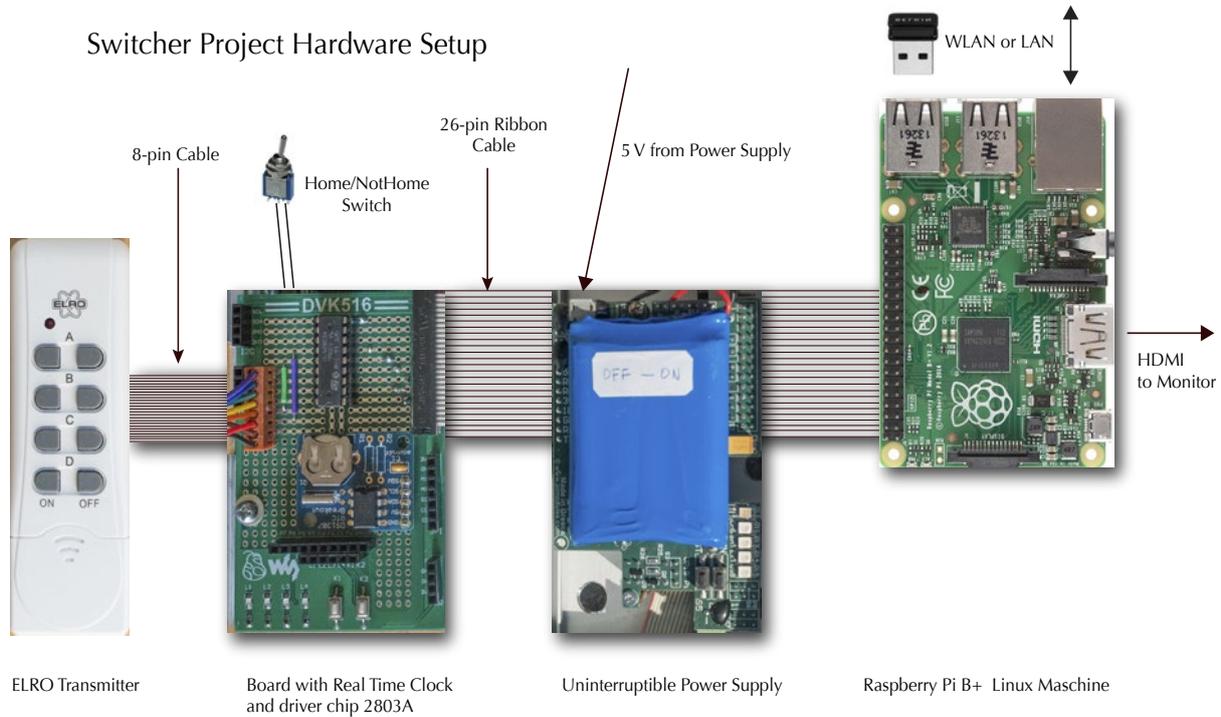


Links: Pi stacked mit Schaltboard - Rechts: UPiS mit Accu



Switcher komplett

5. Hardware Aufbau



Hardware Setup Switcher

Der Pi, resp. das ganze System, wird via das uninterruptible Power Supply UPiS (mit blauen LiPO-Accu) mit 5 V Spannung versorgt. Da im ELRO Handsender bereits ein Sende-Protokoll-Chip PT2262 sowie ein 433 MHz Sender enthalten ist, wird der Handsender via ein Treiber-Chip 2803A von den General Purpose Input/Output Pins (GPIO-Pins) des Pi angesteuert.

Das grüne Board enthält eine batteriegepufferte Real Time Clock sowie den Treiber-Chip 2803A. Ebenfalls auf diesem Board befinden sich 3 LED und 2 Drucktasten. LED1 und LED3 werden für Statusanzeigen benutzt. Die Drucktasten sind zZ. nicht benutzt.

Das Board enthält ebenfalls 2 Anschlüsse für einen im Gehäuse montierten Mini-Kippschalter. Dieser hat 2 Positionen: ‚Nobody Home‘ und ‚Somebody Home‘. Siehe entsprechende Kapitel.

Die Boards wurden in einem handelsüblichen Alu-Gehäuse montiert.

5.1 Pin Belegung Raspberry Pi

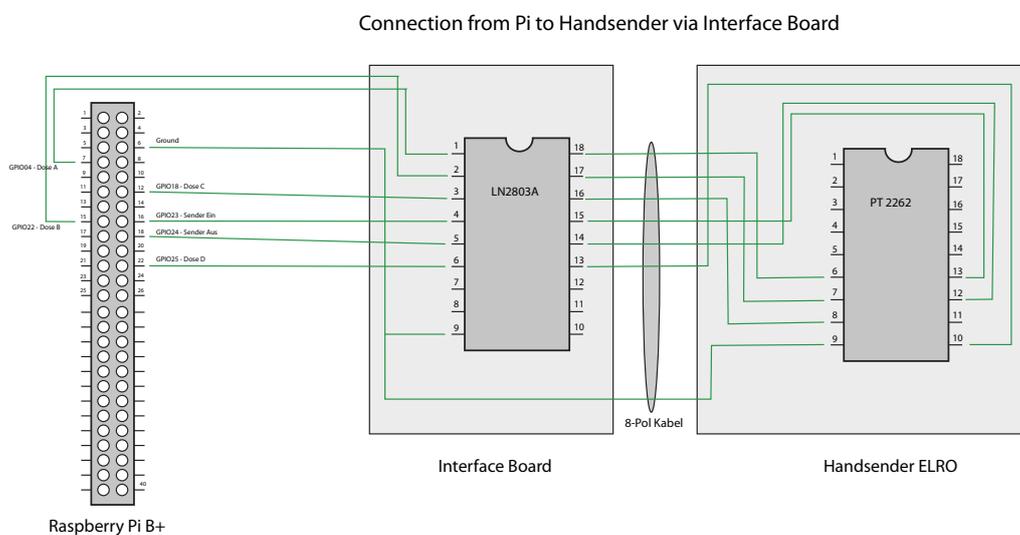
Der verwendete Raspberry Pi Modell B+ hat 40-Pins, es werden jedoch nur die ersten 26 Pins verwendet.

Pin	GPIO	Richtung	Verwendung
3	2		I2C Bus
5	3		i2C Bus
7	04	OUT	Schaltdose A
8	14		Serial Interface Tx
10	15		Serial Interface Rx
11	17		Relais UPiS
12	18	OUT	Schaltdose C
13	27	IN	Soft Shut Down
15	22	OUT	Schaltdose B
16	23	OUT	Sender ausschalten
18	24	OUT	Sender einschalten

19	10	OUT	LED 1 - Switcher läuft
21	9	OUT	LED 2 (not used)
22	25	OUT	Schaltdose D
23	11	OUT	LED 3 (Script pi_running.py)
24	8	IN	Druckschalter 1
26	7		Home/Not Home Switch

5.2 Verbindung zum Handsender

Dieses Schema zeigt die Verbindungen vom Pi zum Chip (auf Interface Board) und zum Handsender.



5.3 UPiS

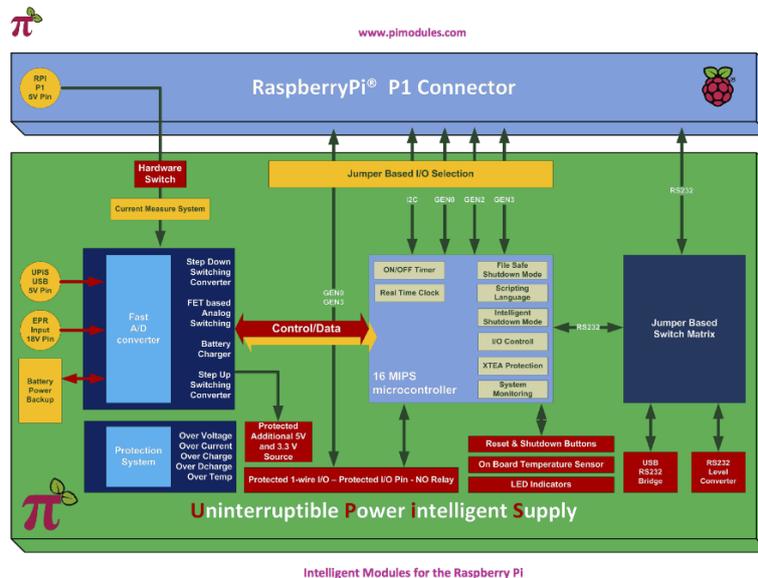
Das verwendete Un-interruptible Power Supply Module (genannt UPiS) mit dem blauen LiPo-Accu wird von einer in Athen, Griechenland, beheimateten Firma hergestellt.

Note:

In der Zwischenzeit erfahren: diese Firma ist nicht zuverlässig, das Produkt wird nicht mehr gewartet und ich täte diese Firma nicht mehr berücksichtigen in zukünftigen Projekten.

Das Module hat vielfältige Funktionen, es kann weitgehend konfiguriert werden mittels eines seriellen Interfaces, welches über die Pins 8 und 10 des Pi läuft. Fällt die von Aussen kommende 5 Volt Stromversorgung aus, so wird das System während ca. 3-6 Stunden ab dem Accu mit Strom versorgt. Wird der untere Ladezustand erreicht, so wird der Pi kontrolliert heruntergefahren.

Über das serielle Interface können Commands an das UPiS gesendet werden, einer dacon, der Command @status bringt den internen Status des UPiS.



Blockschema des UPIs

5.4 Nobody Home Switch

Der Switcher ist nicht nur während Ferienabwesenheiten in Betrieb, sondern oft auch beim normalen Daheim-Sein. Da kommt es vor, dass man Abends weg ist, resp. im Laufe des Abends heimkommt. Mit dem Kippschalter Home/NotHome kann die Funktion des Switchers folgendermassen beeinflusst werden:

- Stellung NotHome: Switcher schaltet die Dosen gemäss den Angaben in Control-File
- Stellung Home: Switcher schaltet nach Umschalten auf Home alle Dosen sofort AUS. Folgende Aktionen (Aus/Ein) werden nicht mehr durchgeführt, der aktuelle Dosenstatus wird jedoch nachgeführt.

Beim Umschalten von Home auf NoHome werden die Dosen sofort wieder auf den zu diesem Zeitpunkt gültigen Stand geschaltet. Das vordefinierte Programm wird wieder normal abgearbeitet.

6. Software

6.1 Switcher-Programm

Die verwendete Script-Sprache ist Python. Das Switcher-Programm **switcher.py** ist ca. 550 Zeilen lang, es steuert den Handsender (und damit auf Distanz die Funksteckdosen) nach einem festgelegten Wochenprogramm.

Die Schaltsequenzen sind extern des Programms in einem XML-File codiert. Dieser File wird nach Programm-Start eingelesen und geparkt und es entsteht eine Aktionen-Tabelle, welche alle Schaltvorgänge (EIN/AUS) für jeden Tag der Woche und für jede Dose enthält. Es können beliebig viele Schaltvorgänge pro Dose und Tag definiert werden. Das Wochenprogramm wird endlos wiederholt.

Das Switcher-Programm enthält eine Simulations-Funktion (Commandline Parameter -s), welche einen Schnelldurchlauf durch 3 Wochen in ca. 10 Minuten ermöglicht. Damit kann das korrekte Funktionieren des Programms im Wochenablauf schnell überprüft werden.

Das Switcher-Programm kann auf zwei Arten gestartet werden:

- Einfacher Aufruf auf Commandline: `sudo python switcher.py [- Commandline Parameter]`

- Aufruf als Linux-Daemon, siehe Kapitel weiter hinten

Beim Start des Programms können folgenden Commandline-Parameter spezifiziert werden:

```
-d          kleiner debug, Statusmeldungen werden ausgegeben (stdout)
-D          grosser debug, weitere Statusmeldungen
-f filename XML Inputfile
-s          Simulation (Schnelldurchlauf durch 3 wochen
-i min     Inkrement-Intervall bei Simulation (Minuten, 2-8 sinnvoll)
-t wochentag bei dem die Simulation gestartet wird
-h          help, usage wird ausgegeben
-a          es werden nur die im File gefundenen Aktionen ausgegeben
```

6.2 Control File

Der Control-File in XML Format beinhaltet alle Schaltsequenzen für alle Tage der Woche pro Dose. Der File hat im Tag <file-id> eine Beschreibung - damit kann der aktuell gültige File identifiziert werden. Siehe Statusmeldung des Switcher-Clients. Der Control-File wird mit einem Editor erstellt. Es ist dabei sorgfältig vorzugehen - Fehler in den Tags sind leicht zu übersehen und führen zu einem Absturz des Switcher Programms.

Hier ein Ausschnitt des Files:

```
<control>
<file-id>4 Dosen winter bei Peter</file-id>
<!--
Dose Nummer 1 wohnzimmer***** Dose 1
-->
<dose>
<dose-nr>
  1
</dose-nr>
<day>
<weekday>0</weekday>
  <sequence>
    <start>00.10</start>
    <stop>00.40</stop>
  </sequence>
  <sequence>
    <start>18.00</start>
    <stop>22.00</stop>
  </sequence>
  <sequence>
    <start>22.45</start>
    <stop>23.15</stop>
  </sequence>
</day>
<day>
<weekday>1</weekday>
  <sequence>
```

(...)

6.3 Switcher Client

Das Switcher-Client Programm **swclient.py** kann Request an den Switcher Daemon senden. Siehe folgendes Kapitel Interprocess Kommunikation.

Der Status-Request bringt folgende Antwort vom Daemon:

```
pi@raspberrypi ~/switcher $ sudo python swclient.py
Please give a command: stat
Sending: stat
-----
```

```
* Status: 2
* Running since: Friday 23.01.2015 08.20
* XML: /home/pi/switcher/control.xml
* File-ID: 4 Dosen winter bei Peter
* Logging: 0
* Debug: 0
* Weekday: 4
* AnzActions: 228
* Dosenstatus: 0000
* WaitFor: 17.38
* ZeitNow: 07.43
* LastAction: Zeit: 00.22 Dose: 1 / OFF
* NextAction: Zeit: 17.38 Dose: 3 / ON
* Home: Nobody Home
```

Please give a command:

In diesem Beispiel ist zu sehen, dass der Switcher-Daemon auf Zeit 17.38 Uhr wartet, um dann die Dose 3 einzuschalten. Letzte Aktion war um 0.22 Uhr, als Dose 1 ausgeschaltet wurde. Alle Dosen sind zur Zeit ausgeschaltet (Dosenstatus). Der Home/NotHome Switch ist in Stellung NotHome.

6.4 Schaltzeiten anzeigen

Das Script **swaction.py** wird in der Entwicklung benutzt, um die im XML-File codierten Schaltzeiten pro Tag und pro Dose auszugeben - für Tests und Doku. Es werden 3 Listen ausgegeben - [siehe Beispiel im Anhang](#).

- Schaltzeiten für alle Tage der Woche für alle Dosen.
- Schaltzeiten für jede Dose für alle Tage der Woche
- Grafische Darstellung (Timeline) für alle Tage und Dosen

Dieses Script benutzt dieselbe Parsing-Funktion wie das Switcher-Script switcher.py. Diese Parsing-Funktion ist im Script swparse.py zu finden.

Dieselben Listen werden auch vom Switcher Script switcher.py ausgegeben, falls der Commandine Parameter -a gesetzt ist.

Für Ausgabe in einen Textfile ist folgender Command einzugeben:

```
sudo python swaction.py > schaltzeiten.txt
```

7. Daemonisierung

Das Switcher-Programm switcher.py erfüllt alle Anforderungen des Schaltens der Dosen.

7.1 Switcher mit Daemon

Als weitere Herausforderung wurde der Switcher-Prozess ‚daemonisiert‘, d.h. es wurden weitere Code-Teile angefügt, die es erlauben, den Switcher als Linux-Daemon laufen zu lassen. Der Switcher-Daemon kann von der Commandline mit folgenden Aufrufen gestartet und gestoppt werden:

```
sudo python swdaem.py start
```

```
sudo python swdaem.py stop
```

Der dafür zusätzlich notwendige Code wurde hier gefunden und leicht angepasst:

[Linux Daemon in Python](#)

7.2 Switcher ohne Daemon

Der Switcher Code kann auch ohne Daemon ausgeführt werden. Linux Daemons sind nicht jedermanns Sache und deshalb ist der Code so gebaut, dass das Script (Hauptscript) **switcher.py** auch für sich allein aufgerufen werden kann - mit den genannten Commandline Parameter.

In diesem Fall wird das Script `swdaem.py` einfach weggelassen - es wird nicht verwendet.

Der Switcher wird dann so gestartet (aus dem Folder `switcher`)

```
sudo python switcher.py
```

In der Datei `/etc/rc.local` kann in diesem Fall folgende Zeile eingefügt werden, um den Switcher beim boot zu starten:

```
# switcher start
python /home/pi/switcher/switcher.py &
```

Damit wird auch keine pid Datei erstellt und der entsprechende Code wird aus `/etc/rc.local` entfernt. Die Interprocess Kommunikation bleibt dabei bestehen, sie wird ja im Script `swipc.py` abgehandelt. Das Client-Script ist `swclient.py`

8. Interprocess Kommunikation

Da der Switcher-Prozess als Linux Daemon im Hintergrund läuft und damit keine Verbindung zu einer Konsole hat, ist das Switcher-Programm als Server ausgeführt, welches neben dem Schalten der Dosen auch Requests von einem Client beantworten kann. Eine dieser Requests ist der „stat“-request, der etwa folgendes meint: sag mir, was du jetzt gerade tust, was du als letztes getan hast und was du als nächstes tun wirst.

Diese Interprocess-Kommunikation wird über Sockets abgewickelt - es wird dazu das geniale Framework ZeroMQ verwendet: ‚an intelligent transport layer for distributed applications‘.

Aus dem Guide:

ØMQ (also known as ZeroMQ, 0MQ, or zmq) looks like an embeddable networking library but acts like a concurrency framework. It gives you sockets that carry atomic messages across various transports like in-process, inter-process, TCP, and multicast. You can connect sockets N-to-N with patterns like fan-out, pub-sub, task distribution, and request-reply. It's fast enough to be the fabric for clustered products. Its asynchronous I/O model gives you scalable multicore applications, built as asynchronous message-processing tasks. It has a score of language APIs and runs on most operating systems. ØMQ is from iMatix and is LGPLv3 open source.

Das Switcher-Programm fragt im Main-Loop nonblocking die Queue ab und kann so Anfragen vom Client-Programm **swclient.py** beantworten.

Das Client-Programm **swclient.py** wird bei Bedarf in einer Konsole gestartet.

Eine Status-Abfrage ist zudem auch via ein Web-Interface möglich - siehe weiter hinten. Aus diesem Grund ist im Pi der Web Server `lighttpd` installiert.

9. Shutdown Pi

Für ordentlichen Shutdown des Pi läuft das Script **softshut_upis.py**. Dieses prüft, ob Pin GPIO27 auf Null geht - bei Tastendruck (rote Taste) auf den UPiS Board. Der Pi wird dann heruntergefahren mit dem Befehl halt. Das UPiS sorgt dann dafür, dass die 5V Versorgung des Pi abgeschaltet wird. Druck auf denselben Knopf schaltet 5V zum Pi wieder ein.

10. Web-Interface

Auf dem Pi des Switchers ist der Web-Server lighttpd incl. PHP installiert. Via Port 8008 kann der Status des Switchers analog dem Commandline Client **swclient.py** abgefragt werden. Mittels Port-Forwarding kann dieser auch von ausserhalb des lokalen Netzwerks angefragt werden. Die gelieferte Webseite sieht so aus:



Statusabfrage via Web

11. Software Komponenten

Das Switcher-Projekt besteht aus folgenden Codeteilen:

Script Name	Funktion
switcher.py	Hauptprogramm Switcher, schaltet die Dosen
switcher_set.py	Stand alone Switcher (needs cmd-line parameter) for Web-Interface
swclient.py	Switcher-Client, sendet requests an den Switcher-Server
swipc.py	Server-Komponente des Switcher Scripts
swparse.py	Module mit den Funktionen parse und print_actions
swdaem.py	Spezifischer Code für Daemonisierung
daemon.py	Allgemeiner Code für Daemonisierung
debug_to_file.py	Komponente von swdaem.py
defglobal.py	Include mit Definition globaler Variablen
control.xml	XML mit den Schaltsequenzen
pi-running.py	Pi background prozess blinkt LED 3 (GPIO11)

softshut_upis.py	Prüft GPIO27 und leitet kontrollierten Shutdown des Pi ein (Druckschalter auf UPiS Board).
swaction.py	Parse xml and print actions - do not more
testswitch.py	Test-Programm, schaltet Dosen ein/aus in Sequenz, checks wiring.

12. Links

Meine Raspberry Projekte

Projekt Xmas TV, Raspeberry Projekt:

[Projekt Xmas TV](#)

Switcher Projekt, Beschreibung und Code auf meiner Dropbox:

[Projekt Switcher](#)

Light Painting Projekt, Bau eines Pixelsticks, Beschreibung und Code siehe Dropbox

[Projekt Light Paint](#)

Andere Links

Erste Idee für die Ansteuerung des Handsenders in einem Artikel von Hermann Kurz gefunden. Er verwendete den ELRO Handsender, da dieser bereits den Sende-Protokoll-Chip PT2262 sowie einen 433 MHz Sender enthält.

<http://www.heise.de/hardware-hacks/projekte/Funksteckdose-im-Internet-1814688.html>

Interprocess-Kommunikation mittels ZeroQM Framework

<http://zeromq.org>

UPiS Module von PI-Modules.com (forget this company !)

<http://www.pimodules.com>

Real Time Clock Module mit Batterie von Adafruit

<http://www.adafruit.com/product/264>

Funksteckdosen-Set von ELRO, Artikelnummer AB440S/3, 3 Stück mit Handsender

<http://www.elro.eu/de/produkte/cat/home-automation/heimautomation/sets1>

Beste Firma für Ideen und Anleitungen

<https://www.adafruit.com>

In der CH zu empfehlen, führt Adafruit Produkte

<http://www.play-zone.ch>

13. Anhang

Hier der Output des Scripts swaction.py

XML file found: /home/pi/switcher/control.xml
Control-File-ID : 4 Dosen Sommer bei Peter

Liste 1: Aktionen pro Wochentag -----

Wochentag: Sonntag - Anzahl Aktionen: 14

Zeit: 00.10 Dose: 1 Switch On
Zeit: 00.30 Dose: 1 Switch Off
Zeit: 19.55 Dose: 3 Switch On
Zeit: 20.35 Dose: 2 Switch On
Zeit: 20.45 Dose: 1 Switch On
Zeit: 21.12 Dose: 3 Switch Off
Zeit: 21.15 Dose: 2 Switch Off
Zeit: 22.00 Dose: 1 Switch Off
Zeit: 22.10 Dose: 2 Switch On
Zeit: 22.45 Dose: 1 Switch On
Zeit: 22.45 Dose: 2 Switch Off
Zeit: 22.52 Dose: 4 Switch On
Zeit: 23.04 Dose: 4 Switch Off
Zeit: 23.15 Dose: 1 Switch Off

Wochentag: Montag - Anzahl Aktionen: 18

Zeit: 00.15 Dose: 1 Switch On
Zeit: 00.25 Dose: 1 Switch Off
Zeit: 20.15 Dose: 2 Switch On
Zeit: 20.17 Dose: 3 Switch On
Zeit: 21.05 Dose: 1 Switch On
Zeit: 21.14 Dose: 2 Switch Off
Zeit: 21.30 Dose: 3 Switch Off
Zeit: 22.17 Dose: 4 Switch On
Zeit: 22.20 Dose: 1 Switch Off
Zeit: 22.29 Dose: 4 Switch Off
Zeit: 23.05 Dose: 1 Switch On
Zeit: 23.18 Dose: 3 Switch On
Zeit: 23.18 Dose: 4 Switch On
Zeit: 23.20 Dose: 2 Switch On
Zeit: 23.30 Dose: 4 Switch Off
Zeit: 23.40 Dose: 2 Switch Off
Zeit: 23.42 Dose: 1 Switch Off
Zeit: 23.47 Dose: 3 Switch Off

Wochentag: Dienstag - Anzahl Aktionen: 16

Zeit: 19.50 Dose: 2 Switch On
Zeit: 20.40 Dose: 3 Switch On
Zeit: 20.55 Dose: 1 Switch On
Zeit: 21.15 Dose: 2 Switch Off
Zeit: 21.45 Dose: 1 Switch Off
Zeit: 21.45 Dose: 3 Switch Off
Zeit: 21.55 Dose: 4 Switch On
Zeit: 22.15 Dose: 4 Switch Off
Zeit: 22.20 Dose: 1 Switch On
Zeit: 23.22 Dose: 1 Switch Off
Zeit: 23.36 Dose: 3 Switch On
Zeit: 23.36 Dose: 4 Switch On
Zeit: 23.45 Dose: 2 Switch On
Zeit: 23.46 Dose: 4 Switch Off
Zeit: 23.56 Dose: 2 Switch Off
Zeit: 23.56 Dose: 3 Switch Off

Wochentag: Mittwoch - Anzahl Aktionen: 24

Zeit: 00.02 Dose: 1 Switch On
Zeit: 00.05 Dose: 2 Switch On
Zeit: 00.09 Dose: 3 Switch On
Zeit: 00.22 Dose: 1 Switch Off
Zeit: 00.24 Dose: 2 Switch Off
Zeit: 00.29 Dose: 3 Switch Off
Zeit: 01.30 Dose: 4 Switch On
Zeit: 01.38 Dose: 4 Switch Off
Zeit: 20.45 Dose: 2 Switch On
Zeit: 20.50 Dose: 1 Switch On

Zeit: 20.50 Dose: 3 Switch On
Zeit: 21.40 Dose: 2 Switch Off
Zeit: 22.02 Dose: 3 Switch Off
Zeit: 22.05 Dose: 1 Switch Off
Zeit: 22.16 Dose: 2 Switch On
Zeit: 22.25 Dose: 4 Switch On
Zeit: 22.36 Dose: 2 Switch Off
Zeit: 22.41 Dose: 4 Switch Off
Zeit: 22.55 Dose: 1 Switch On
Zeit: 22.56 Dose: 4 Switch On
Zeit: 23.08 Dose: 4 Switch Off
Zeit: 23.30 Dose: 3 Switch On
Zeit: 23.40 Dose: 1 Switch Off
Zeit: 23.49 Dose: 3 Switch Off

Wochentag: Donnerstag - Anzahl Aktionen: 18

Zeit: 00.02 Dose: 1 Switch On
Zeit: 00.22 Dose: 1 Switch Off
Zeit: 20.05 Dose: 3 Switch On
Zeit: 20.45 Dose: 2 Switch On
Zeit: 21.28 Dose: 2 Switch Off
Zeit: 21.30 Dose: 1 Switch On
Zeit: 22.00 Dose: 3 Switch Off
Zeit: 22.30 Dose: 4 Switch On
Zeit: 22.35 Dose: 3 Switch On
Zeit: 22.40 Dose: 1 Switch Off
Zeit: 22.46 Dose: 4 Switch Off
Zeit: 22.47 Dose: 2 Switch On
Zeit: 22.55 Dose: 1 Switch On
Zeit: 23.10 Dose: 2 Switch Off
Zeit: 23.14 Dose: 3 Switch Off
Zeit: 23.20 Dose: 1 Switch Off
Zeit: 23.53 Dose: 4 Switch On
Zeit: 23.58 Dose: 4 Switch Off

Wochentag: Freitag - Anzahl Aktionen: 24

Zeit: 00.012 Dose: 2 Switch On
Zeit: 00.07 Dose: 1 Switch On
Zeit: 00.10 Dose: 3 Switch On
Zeit: 00.10 Dose: 4 Switch On
Zeit: 00.17 Dose: 1 Switch Off
Zeit: 00.17 Dose: 4 Switch Off
Zeit: 00.27 Dose: 2 Switch Off
Zeit: 00.37 Dose: 3 Switch Off
Zeit: 20.30 Dose: 2 Switch On
Zeit: 20.47 Dose: 3 Switch On
Zeit: 21.00 Dose: 1 Switch On
Zeit: 21.25 Dose: 2 Switch Off
Zeit: 21.27 Dose: 3 Switch Off
Zeit: 21.40 Dose: 1 Switch Off
Zeit: 21.47 Dose: 4 Switch On
Zeit: 21.57 Dose: 4 Switch Off
Zeit: 22.00 Dose: 1 Switch On
Zeit: 22.15 Dose: 2 Switch On
Zeit: 22.35 Dose: 2 Switch Off
Zeit: 23.15 Dose: 1 Switch Off
Zeit: 23.15 Dose: 3 Switch On
Zeit: 23.15 Dose: 4 Switch On
Zeit: 23.21 Dose: 4 Switch Off
Zeit: 23.34 Dose: 3 Switch Off

Wochentag: Samstag - Anzahl Aktionen: 12

Zeit: 00.02 Dose: 1 Switch On
Zeit: 00.22 Dose: 1 Switch Off
Zeit: 20.25 Dose: 1 Switch On
Zeit: 20.25 Dose: 2 Switch On
Zeit: 20.37 Dose: 3 Switch On
Zeit: 21.35 Dose: 3 Switch Off
Zeit: 21.45 Dose: 2 Switch Off
Zeit: 22.05 Dose: 1 Switch Off
Zeit: 23.07 Dose: 4 Switch On
Zeit: 23.10 Dose: 1 Switch On
Zeit: 23.16 Dose: 4 Switch Off
Zeit: 23.40 Dose: 1 Switch Off

End Aktionen pro Wochentag -----

Liste 2: Aktionen pro Dose -----

```

---- Sonntag -----
---- Dose 1, Anzahl Actions 6
Zeit: 00.10, Action: ON
Zeit: 00.30, Action: OFF
Zeit: 20.45, Action: ON
Zeit: 22.00, Action: OFF
Zeit: 22.45, Action: ON
Zeit: 23.15, Action: OFF
---- Dose 2, Anzahl Actions 4
Zeit: 20.35, Action: ON
Zeit: 21.15, Action: OFF
Zeit: 22.10, Action: ON
Zeit: 22.45, Action: OFF
---- Dose 3, Anzahl Actions 2
Zeit: 19.55, Action: ON
Zeit: 21.12, Action: OFF
---- Dose 4, Anzahl Actions 2
Zeit: 22.52, Action: ON
Zeit: 23.04, Action: OFF
---- Montag -----
---- Dose 1, Anzahl Actions 6
Zeit: 00.15, Action: ON
Zeit: 00.25, Action: OFF
Zeit: 21.05, Action: ON
Zeit: 22.20, Action: OFF
Zeit: 23.05, Action: ON
Zeit: 23.42, Action: OFF
---- Dose 2, Anzahl Actions 4
Zeit: 20.15, Action: ON
Zeit: 21.14, Action: OFF
Zeit: 23.20, Action: ON
Zeit: 23.40, Action: OFF
---- Dose 3, Anzahl Actions 4
Zeit: 20.17, Action: ON
Zeit: 21.30, Action: OFF
Zeit: 23.18, Action: ON
Zeit: 23.47, Action: OFF
---- Dose 4, Anzahl Actions 4
Zeit: 22.17, Action: ON
Zeit: 22.29, Action: OFF
Zeit: 23.18, Action: ON
Zeit: 23.30, Action: OFF
---- Dienstag -----
---- Dose 1, Anzahl Actions 4
Zeit: 20.55, Action: ON
Zeit: 21.45, Action: OFF
Zeit: 22.20, Action: ON
Zeit: 23.22, Action: OFF
---- Dose 2, Anzahl Actions 4
Zeit: 19.50, Action: ON
Zeit: 21.15, Action: OFF
Zeit: 23.45, Action: ON
Zeit: 23.56, Action: OFF
---- Dose 3, Anzahl Actions 4
Zeit: 20.40, Action: ON
Zeit: 21.45, Action: OFF
Zeit: 23.36, Action: ON
Zeit: 23.56, Action: OFF
---- Dose 4, Anzahl Actions 4
Zeit: 21.55, Action: ON
Zeit: 22.15, Action: OFF
Zeit: 23.36, Action: ON
Zeit: 23.46, Action: OFF
---- Mittwoch -----
---- Dose 1, Anzahl Actions 6
Zeit: 00.02, Action: ON
Zeit: 00.22, Action: OFF
Zeit: 20.50, Action: ON
Zeit: 22.05, Action: OFF
Zeit: 22.55, Action: ON
Zeit: 23.40, Action: OFF
---- Dose 2, Anzahl Actions 6
Zeit: 00.05, Action: ON

```

```

Zeit: 00.24, Action: OFF
Zeit: 20.45, Action: ON
Zeit: 21.40, Action: OFF
Zeit: 22.16, Action: ON
Zeit: 22.36, Action: OFF
---- Dose 3, Anzahl Actions 6
Zeit: 00.09, Action: ON
Zeit: 00.29, Action: OFF
Zeit: 20.50, Action: ON
Zeit: 22.02, Action: OFF
Zeit: 23.30, Action: ON
Zeit: 23.49, Action: OFF
---- Dose 4, Anzahl Actions 6
Zeit: 01.30, Action: ON
Zeit: 01.38, Action: OFF
Zeit: 22.25, Action: ON
Zeit: 22.41, Action: OFF
Zeit: 22.56, Action: ON
Zeit: 23.08, Action: OFF
---- Donnerstag -----
---- Dose 1, Anzahl Actions 6
Zeit: 00.02, Action: ON
Zeit: 00.22, Action: OFF
Zeit: 21.30, Action: ON
Zeit: 22.40, Action: OFF
Zeit: 22.55, Action: ON
Zeit: 23.20, Action: OFF
---- Dose 2, Anzahl Actions 4
Zeit: 20.45, Action: ON
Zeit: 21.28, Action: OFF
Zeit: 22.47, Action: ON
Zeit: 23.10, Action: OFF
---- Dose 3, Anzahl Actions 4
Zeit: 20.05, Action: ON
Zeit: 22.00, Action: OFF
Zeit: 22.35, Action: ON
Zeit: 23.14, Action: OFF
---- Dose 4, Anzahl Actions 4
Zeit: 22.30, Action: ON
Zeit: 22.46, Action: OFF
Zeit: 23.53, Action: ON
Zeit: 23.58, Action: OFF
---- Freitag -----
---- Dose 1, Anzahl Actions 6
Zeit: 00.07, Action: ON
Zeit: 00.17, Action: OFF
Zeit: 21.00, Action: ON
Zeit: 21.40, Action: OFF
Zeit: 22.00, Action: ON
Zeit: 23.15, Action: OFF
---- Dose 2, Anzahl Actions 6
Zeit: 00.012, Action: ON
Zeit: 00.27, Action: OFF
Zeit: 20.30, Action: ON
Zeit: 21.25, Action: OFF
Zeit: 22.15, Action: ON
Zeit: 22.35, Action: OFF
---- Dose 3, Anzahl Actions 6
Zeit: 00.10, Action: ON
Zeit: 00.37, Action: OFF
Zeit: 20.47, Action: ON
Zeit: 21.27, Action: OFF
Zeit: 23.15, Action: ON
Zeit: 23.34, Action: OFF
---- Dose 4, Anzahl Actions 6
Zeit: 00.10, Action: ON
Zeit: 00.17, Action: OFF
Zeit: 21.47, Action: ON
Zeit: 21.57, Action: OFF
Zeit: 23.15, Action: ON
Zeit: 23.21, Action: OFF
---- Samstag -----
---- Dose 1, Anzahl Actions 6
Zeit: 00.02, Action: ON

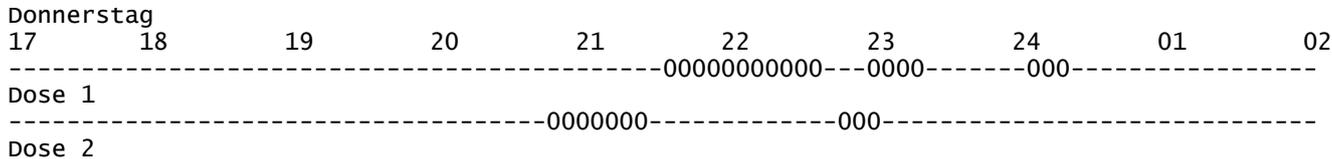
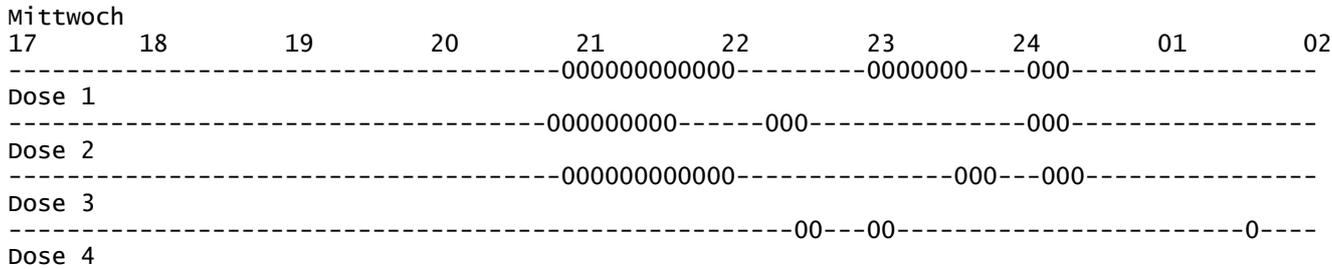
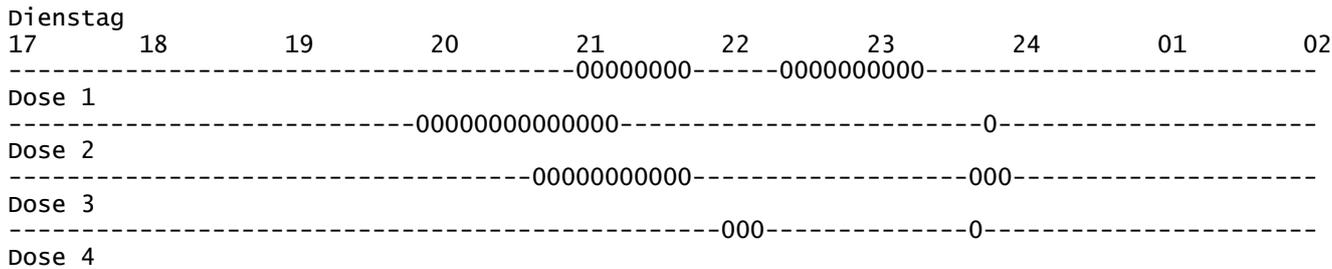
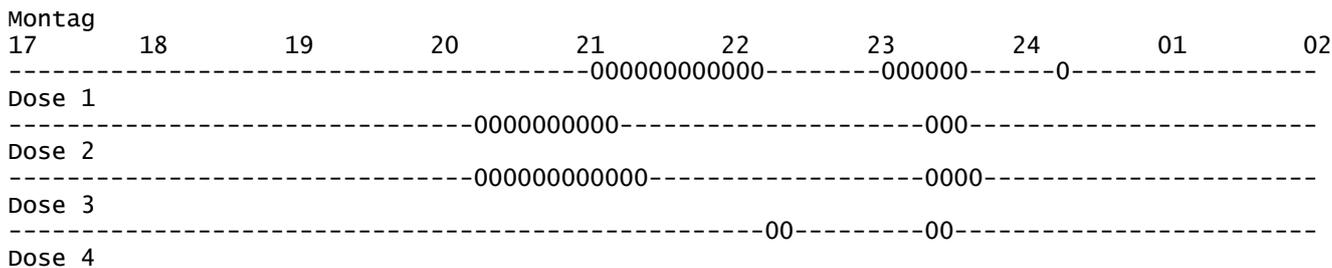
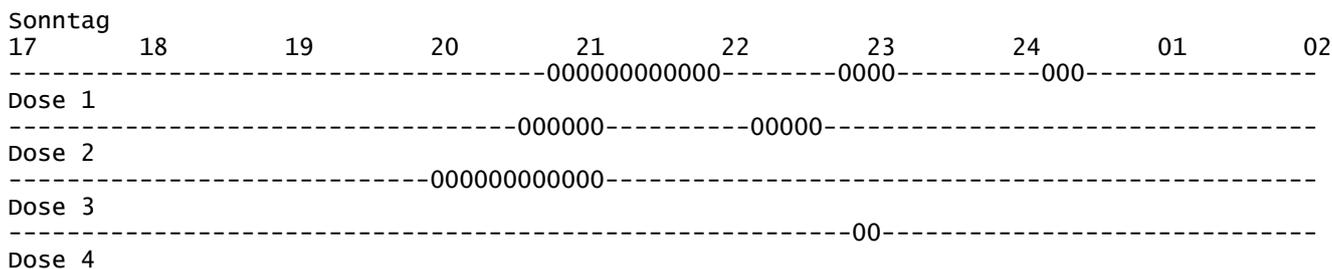
```

```

Zeit: 00.22, Action: OFF
Zeit: 20.25, Action: ON
Zeit: 22.05, Action: OFF
Zeit: 23.10, Action: ON
Zeit: 23.40, Action: OFF
---- Dose 2, Anzahl Actions 2
Zeit: 20.25, Action: ON
Zeit: 21.45, Action: OFF
---- Dose 3, Anzahl Actions 2
Zeit: 20.37, Action: ON
Zeit: 21.35, Action: OFF
---- Dose 4, Anzahl Actions 2
Zeit: 23.07, Action: ON
Zeit: 23.16, Action: OFF
End Aktionen pro Dose -----

```

Liste 3: Schaltaktionen Graphisch -----



```

-----0000000000000000000-----000000-----
Dose 3
-----00-----0-----
Dose 4

```

```

Freitag
17      18      19      20      21      22      23      24      01      02
-----000000-----0000000000000-----0-----
Dose 1
-----0000000000-----000-----0000-----
Dose 2
-----0000000-----000-----0000-----
Dose 3
-----0-----0-----0-----
Dose 4

```

```

Samstag
17      18      19      20      21      22      23      24      01      02
-----00000000000000000-----0000-----000-----
Dose 1
-----00000000000000-----
Dose 2
-----0000000000-----
Dose 3
-----0-----
Dose 4

```

End Aktionen Graphisch -----

Program terminated....

Peter K. Boxler, Oktobr 2014

