Die Ansteuerung der Funktionen per Website

Inhaltsverzeichnis

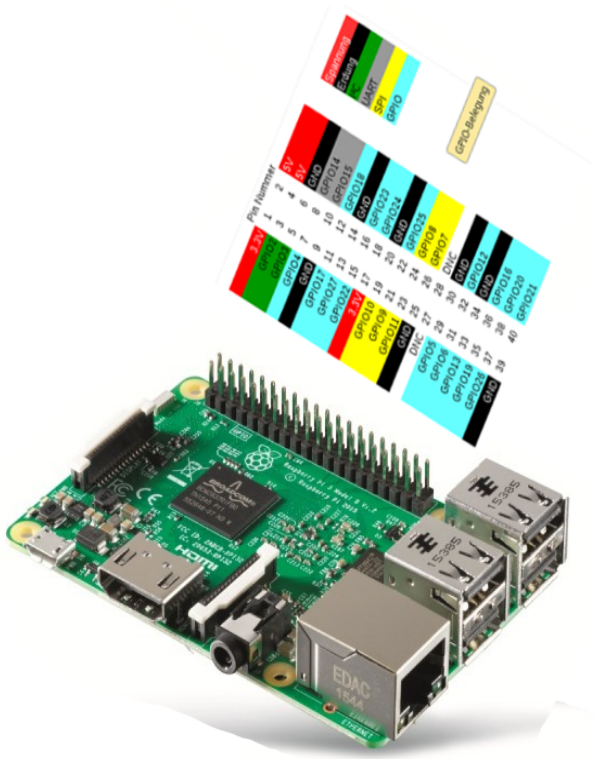
Voraussetzung

- GPIO Steuerung per Python funktioniert

- Passwortgeschützer Bereich auf der Webseite ist eingerichtet

- Webserver läuft und hat root Zugriff auf die Skripte im ./html/pwd Verzeichnis.

1 Nochmal die GPIO-Schnittstelle



WiringPi	Pin-Namen	Pin	Pin	Pin-Namen	WiringPi
–	+ 3,3 V	**1**	**2**	+ 5 V	–
8	(SDA1) GPIO 2	**3**	**4**	+ 5 V	–
9	(SCL1) GPIO 3	**5**	**6**	GND	–
7	(GPIO_GCLK) GPIO 4	**7**	**8**	GPIO 14 (TXD0)	15
–	GND	**9**	**10**	GPIO 15 (RXD0)	16
0	(GPIO_GEN0) GPIO 17	**11**	**12**	GPIO 18 (GPIO_GEN1)	1
2	(GPIO_GEN2) GPIO 27	**13**	**14**	GND	–
3	(GPIO_GEN3) GPIO 22	**15**	**16**	GPIO 23 (GPIO_GEN4)	4
–	+ 3,3 V	**17**	**18**	GPIO 24 (GPIO_GEN5)	5
12	(SPI_MOSI) GPIO 10	**19**	**20**	GND	–
13	(SPI_MISO) GPIO 9	**21**	**22**	GPIO 25 (GPIO_GEN6)	6
14	(SPI_SLCK) GPIO 11	**23**	**24**	GPIO 8 (SPI_CE0_N)	10
–	GND	**25**	**26**	GPIO 7 (SPI_CE1_N)	11
30	(nur für I2C) ID_SD	**27**	**28**	ID_SC (nur für I2C)	31
21	GPIO 5	**29**	**30**	GND	
22	GPIO 6	**31**	**32**	GPIO 12	26
23	GPIO 13	**33**	**34**	GND	
24	GPIO 19	**35**	**36**	GPIO 16	27
25	GPIO 26	**37**	**38**	GPIO 20	28
	GND	**39**	**40**	GPIO 21	29

Die Ansteuerung der GPIO‘s mit ssh und Python wurde im KellerPi Projket schon erstellt.

Nochmal kurz:

Python manuell:

```
root@raspi32:/home/pi32# python
Python 2.7.13 (default, Jan 19 2017, 14:48:08)
[GCC 6.3.0 20170124] on linux2
Type "help", "copyright", "credits" or "license" for more
information.
>>> import RPi.GPIO as GPIO
>>> GPIO.setmode(GPIO.BCM)
>>> RELAIS_1_GPIO = 21
>>> GPIO.setup(RELAIS_1_GPIO, GPIO.OUT)
>>> GPIO.output(RELAIS_1_GPIO, GPIO.HIGH)
>>> GPIO.output(RELAIS_1_GPIO, GPIO.LOW)
```

kann schon mal 3,3V Spannung an Pin 33 ein und ausgeschaltet werden.

Existierende Skrikte:

`./r1-open` (GPIO21 0V; Raspi Stromversorgung aus)

`./r1-close` (GPIO21 3,3V; Raspi Stromversorgung ein)

`./r2-open` (GPIO20 0V; Router Stromversorgung aus)

`./r2-close` (GPIO20 3,3V; Router Stromversorgung aus)

`./r2-reset` (GPIO20 0V für 15s; Router Stromversorgung für 15s aus)

2 Testphase

Da die Hardwareansteuerung root-Rechte benötigt musste ich erst herausfinden, wie man überhaupt Code als root über eine Webseite ausführt ohne gleich den ganzen Raspi per www administrierbar zu machen.

Das Skript:

`root-test-mit-ls.php`

Wird über

```
<div id="Relais">  <a href="root-test-mit-ls.php"><img src="nikolaus.gif"
alt="Relais" > </a> </div>
```

halt mit Click auf das Haus vom Nikolaus aufgerufen und ausgeführt.

Inhalt:

```
<?php
echo "wlan-ein.php";
echo "<br> Inhalt eines Userverzeichnisses:<br>";
echo "shell_exec(''ls /var/www/html/pwd/'');";
echo shell_exec("ls /var/www/html/pwd/");
echo "<br><br><br> nur durch root / www-data lesbar:<br>";
echo "shell_exec(''sudo ls /var/www/html/pwd/'');";
echo shell_exec("sudo ls /var/www/html/pwd/");
echo "<br><br><br> mit ls.py:<br>";
echo "shell_exec(''python /var/www/html/pwd/ls.py'');<br>";
echo shell_exec("python /var/www/html/pwd/ls.py");
echo "<br>";
echo "<br> shell_exec(''sudo python /var/www/html/pwd/ls.py'');<br>";
echo shell_exec("sudo python /var/www/html/pwd/ls.py");
echo "<br>";
echo "<br> shell_exec(''sudo /var/www/html/pwd/ls.sh'');<br>";
echo shell_exec("sudo /var/www/html/pwd/ls.sh");
echo "<br>";
?>
```

Das auflisten der Verzeichnisse dient zum Test, der root-Zugriff direkt auf ein root Verzeichnis dürfte nicht klappen.
Der Raspi/Apache ist so konfiguriert sein, das er nur Skripte/Programme im Verzeichnis ./html/pwd/ als root ausführen kann. Daher brauche ich noch ein Skript, welches als root ausgeführt werden kann und seinerseits python als root ausführt.

`nano ls.sh`

`chmod +x ls.sh` (das gibt Ausführrechte für alle, geht auch)

Inhalt:
`sudo python /var/www/html/pwd/ls.py`

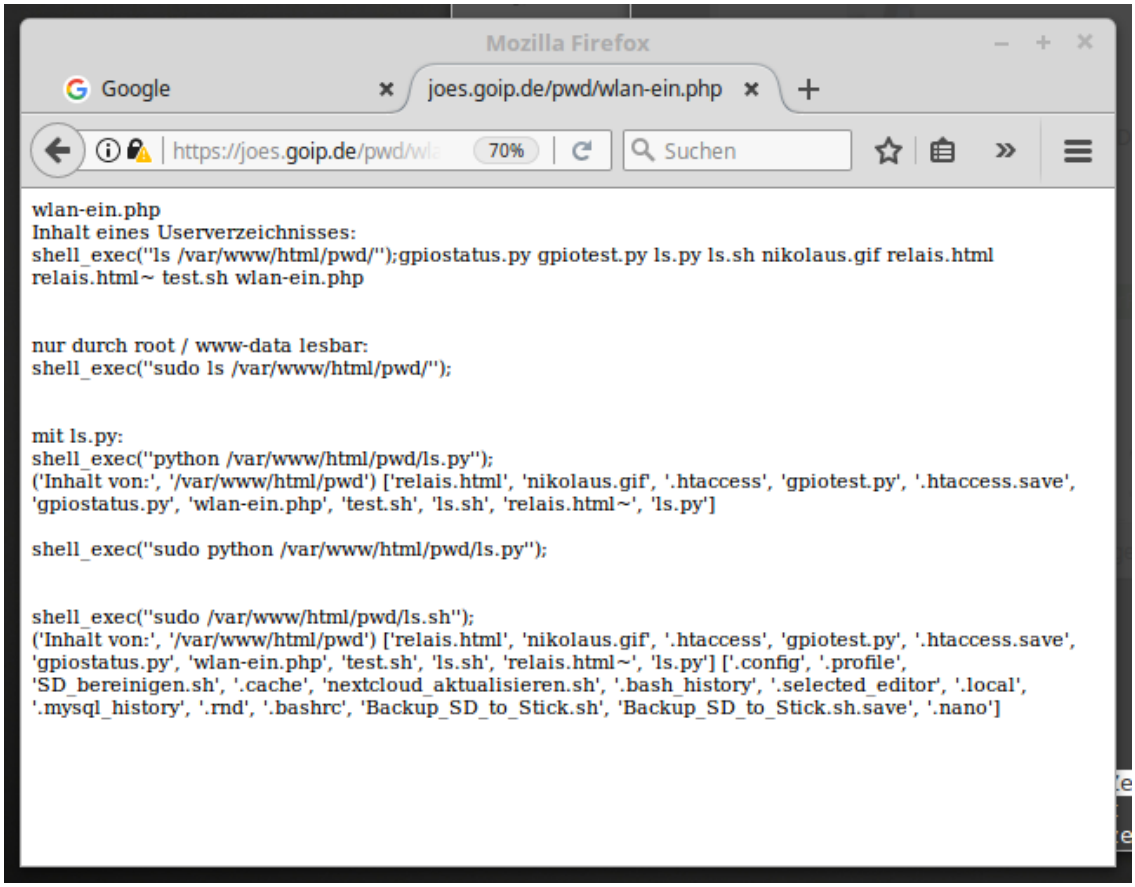
der dann das noch zu erstellende (eigentlich gewollte) Skript ls.py ausführt:

```
import os
a ="/var/www/html/pwd"
print(os.listdir(a))
b ="/root"
print(os.listdir(b))
```

Welches ein „normales Verzeichnis“ auflistet und eins, dass nur über „root-Zugriff“ erreichbar ist.

Funktioniert das Auflisten der Verzeichnisse im Browser sind die Berechtigungen richtig gesetzt. „ls“ muß das root-Verzeichnis nicht listen, ls.py schon.

Es sollte dann im Browser so aussehen:



Wie zu sehen wird das root Verzeichnis nur durch den letzten Aufruf gelistet.

3 Die eigentlichen Funktionen

Derzeit möchte ich per Webseite

- den Router auf dem Dachboden zurücksetzen können (1a)

- eine Steckdose schalten können (1b)

- das Wlan ein und Ausschalten können (2)

3.1 Router zurücksetzten und Steckdose schalten

Zu (1a) wird das Python Skript r2-reset.py verwendet

Zu (1b) wird das Python Skript r1-open.py und r1-close.py verwendet

Die entsprechenden Shell-Skripte sind dann einfach:

```
cp ls.sh router-reset.sh
nano router-reset.sh
```

Kompletten Inhalt löschen und durch:

```
sudo python /var/www/html/pwd/r2-reset.py
```

ersetzten.

```
cp ls.sh Steckdose-aus.sh
nano Steckdose-aus.sh
```

Inhalt:

```
sudo python /var/www/html/pwd/r1-open.py
```

und

```
cp Steckdose-aus.sh Steckdose-ein.sh
nano Steckdose-ein.sh
```

Inhalt:

```
sudo python /var/www/html/pwd/r1-close.py
```

Diese werden dann per php von der Webseite aufgerufen:

```
<?php
echo "Steckdose aus.php";
echo shell_exec("sudo /var/www/html/pwd/Steckdose aus .sh");
?>
<?php
echo "Steckdose ein .php";
echo shell_exec("sudo /var/www/html/pwd/Steckdose ein.sh");
?>
```

3.1.1 Festplatte an der Steckdose

Da zur Zeit eine externe Festplatte an der Steckdose und am Raspi angeschlossen ist und ich diese nicht in eingehängtem Zustand ausschalten möchte wird noch abgefragt ob die Festplatte eingebunden ist.
Dies geschieht in dem die Rückmeldung von „blkid“ oder „lsblk“ in eine Variable geschrieben werden und dann mit stringpos überprüft wird ob der mountpoint der Festplatte zugeordnet (in der Ausgabe enthalten) ist. Ist er das nicht liefert stringpos „false“. „If“ muß also fragen ob „false“ oder „nicht false“ (!==false) gemeldet werden.
Mitder Abfrage für z. Bsp. die UUID können auch die Pictogramme für Ein und Aus gesetzt werden.

3.2 Wlan ein - aus

Das Wlan Interface wird mit „hostapd“ gesteuert. Der Dienst hierzu wurde im Projekt bereits konfiguriert, so das es mit:

```
systemctl start hostapd
systemctl enable hostapd

systemctl status hostapd

systemctl stop hostapd
```

gesteuert werden kann.

Die Steuerung läuft analog per der Kette von Skripten:

```
cp router-reset.sh AP-ein.sh
cp AP-ein.sh AP-status.sh
cp AP-ein.sh AP-aus.sh
```

Jeweils den Inhalt durch das folgende ersetzen:

```
nano AP-ein.sh
     echo "AccessPoint KellerPi ein<br>"
     sudo systemctl start hostapd
     sudo systemctl enable hostapd
nano AP-status.sh
     echo "AccessPoint KellerPi Status<br>"
     echo sudo systemctl status hostapd
nano AP-aus.sh
     echo "AccessPoint KellerPi aus<br>"
     sudo systemctl stop hostapd
```

Die entsprechenden php-Aufrufe analog:

```
cp router-reset.php AP-ein.php
nano AP-ein.php
     <?php
     echo "AP-ein<br>";
     echo shell_exec("sudo /var/www/html/pwd/AP-ein.sh");
     ?>
cp AP-ein.php AP-status.php
nano AP-status.php
     <?php
     echo "AP-status<br>";
     echo shell_exec("sudo /var/www/html/pwd/AP-status.sh");
     ?>
```

```
cp AP-ein.php AP-aus.php
nano AP-aus.php
      <?php
      echo "AP-aus<br>";
      echo shell_exec("sudo /var/www/html/pwd/AP-aus.sh");
      ?>
```

und das ganze in der Webseite verlinken……..

Randbemerkungen:

- Eigentlich ist das setzten GPIO auf Low gar nicht nötig, da das die Initialisierung schon macht.

Wie es aussieht setzt schon "GPIO.setwarnings(False)" den Pin auf Low.

- Bei meiner Netzwerkkonfiguration läuft die Netzwerkverbindung Laptop-Raspi über den Router.

Schaltet man den Router per ssh aus geht die Verbindung verloren und der Router wird nicht wieder eingeschaltet. - Unpraktisch.

Workaround – Den Dlan Adapter als Switch benutzen und Raspi und Laptop direkt verbinden oder den Wlan AP vorher aktivieren und die Verbindung nutzen.

Der Router Reset braucht in Summe ca. 1min bis Router wieder da ist.

- Das Auslesen des GPIO Status eines Ausgangs funktioniert bisher so nicht, da sich der Status ändern kann sobald der GPIO als Eingang definiert wird. Der Versuch wurde abgebrochen, da auch nicht mehr benötigt.

Es soll gpiotest.py aufgerufen werden, welche GPIO 26 für 15s ein- und dann wieder ausschalten soll.

Inhalt von gpiotest.py:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
RELAIS_2_GPIO = 26
GPIO.setup(RELAIS_2_GPIO, GPIO.OUT)
GPIO.output(RELAIS_2_GPIO, GPIO.HIGH)
time.sleep(15)
GPIO.output(RELAIS_2_GPIO, GPIO.LOW)
```

Das Auslesen eines GPIO‘s klappt noch nicht, setzt man mit GPIO.setuop den Chanel auf Mode wird der Status auf „LOW“ gesetzt.

Mit Chanel auf Mode „Aus“ versuchen.
