

Projekt zur Erstellung einer Statusanzeige für die Alarmanlage Lupusec XT2(plus) über einen Raspberry Pi incl. Sprachausgabe und Schaltung von Standard Funksteckdosen

Benötigte Teile für die Statusanzeige, Kosten und Beschaffungslinks:

Raspberry Pi 3 (Pi 1 und 2 funktionieren auch. Hier wird eventuell ein optionaler WLAN-Stick notwendig)
Incl. Netzteil, MicroSD-Karte, Gehäuse und Kühlkörper (Kosten z.Z. 69,99 Euro)

https://www.amazon.de/Vilros-Raspberry-Pi-Complete-Kit-Enthalt/dp/B01DC6MKAQ/ref=sr_1_5?s=computers&ie=UTF8&qid=1498039298&sr=1-5&keywords=raspberry+pi+3

Bauelemente und Test-Equipment (Kosten z.Z. 13,21 Euro)

https://www.amazon.de/Elegoo-Electronic-Breadboard-Kondensator-Potentiometer/dp/B01J79YG8G/ref=sr_1_3?s=computers&ie=UTF8&qid=1498039490&sr=1-3-spons&keywords=raspberry+pi+3&psc=1

USB-Lautsprecher (Kosten z.Z. 6,99 Euro)

https://www.amazon.de/Trust-Lautsprecher-USB-Stromversorgung-Smartphone-schwarz/dp/B00JRW0M32/ref=sr_1_3?ie=UTF8&qid=1498041273&sr=8-3&keywords=usb+lautsprecher

433 Mhz Sender (Kosten z.Z. 4,99 Euro)

https://www.amazon.de/Aukru-Superregeneration-Transmitter-Modul-receiver-module/dp/B00OL93IC/ref=sr_1_cc_3?s=aps&ie=UTF8&qid=1498040001&sr=1-3-catcorr&keywords=433+sender+5v

Brennstuhl Funksteckdosenset (Kosten z.Z. 17,09 Euro)

https://www.amazon.de/Brennenstuhl-Funkschalt-Set-Funksteckdosen-Handsender-Kindersicherung/dp/B001AX8QUM/ref=sr_1_cc_1?s=aps&ie=UTF8&qid=1498040468&sr=1-1-catcorr&keywords=brennenstuhl+funksteckdosen

Die Gesamtkosten belaufen sich auf ca. 110,- Euro.

Installation der Software:

Zunächst ist der Raspberry Pi mit einem Raspbian-Image zu versehen und in Euer Netzwerk einzubinden. Dabei ist es egal ob über WLAN oder LAN. Darauf möchte ich hier nicht weiter eingehen, da es hierzu genügend Anleitungen im Internet gibt. Bindet den Raspberry Pi in Euer Netzwerk ein und aktiviert den SSH-Zugriff.

Das Image findet Ihr hier:

<https://www.raspberrypi.org/downloads/raspbian/>

Ich empfehle die Version mit Pixel, da man hier auch eine GUI zur Bedienung von Einstellungen hat. Die Version ohne Pixel funktioniert aber genauso gut.

Nach der Installation müssen wir uns über SSH mit dem Raspberry Pi verbinden. Hierzu können wir z.B. das Dienstprogramm Putty verwenden.

Folgende Installationen müssen nun durchgeführt werden:

```
sudo apt-get update
sudo apt-get upgrade
# notwendig für 433MHz Sender zum Schalten der Steckdosen
sudo git clone git://github.com/xkonni/raspberry-remote.git
cd raspberry-remote
sudo make send
# Apache Webserver ist notwendig zum Empfang der Anweisungen der Alarmanlage
sudo apt-get install apache2
sudo apt-get install php5
### folgende Datei muss editiert werden:
sudo nano /etc/sudoers
am Ende der Datei folgende Zeile einfügen:
    www-data ALL=(ALL) NOPASSWD: ALL
# notwendig für die Sprachausgabe
sudo apt-get -y install mpg321
sudo chmod o+rw /dev/snd/*
sudo amixer cset numid=3 1
sudo amixer set PCM -- 100%
```

Aufbau der Hardware:

Dies ist ein Beispiel-Aufbau, so wie er bei mir im Einsatz ist.

Ich setze insgesamt 6 LED's ein. **Hier bitte unbedingt einen Vorwiderstand verwenden!**

3 LED's zeigen den Status der Anlage an, die anderen 3 LED's signalisieren mir, ob in den (drei) Etagen unseres Hauses Fenster geöffnet sind. (Status ist „disarm“ und im Obergeschoß ist mindestens ein Fenster geöffnet)



Die Beschaltung der GPIO-Ports am Raspberry Pi sieht wie folgt aus:

	Bez.	Pin	Bez.	
	3V3	1	2	5V
	GPIO02	3	4	5V
	GPIO03	5	6	GND
LED gelb DG	GPIO04	7	8	GPIO14
GND Fenster	GND	9	10	GPIO15
433 MHz	GPIO17	11	12	GPIO18
LED gelb OG	GPIO27	13	14	GND
LED gelb EG	GPIO22	15	16	GPIO23
	3V3	17	18	GPIO24
	GPIO10	19	20	GND
	GPIO09	21	22	GPIO25
	GPIO11	23	24	GPIO08
	GND	25	26	GPIO07

Für den 433MHz Sensor ist kein Vorwiderstand notwendig, allerdings eine etwa 17cm lange Drahtantenne an dem dafür vorgesehenen PIN am Modul. Die Lautsprecher einfach an einen freien USB-Port des Raspberry Pi's stecken. Treiber sind hierfür nicht erforderlich.

Konfiguration:

Nun geht es an die individuelle Konfiguration.

Ihr müsst nun php-Dateien erstellen, die in den Webserver eingebunden werden und mittels Action-URL von der Alarmanlage aufgerufen werden können.

Ein Beispiel für disarm:

Wird die Alarmanlage auf „disarm“ gesetzt erfolgt durch den Action-URL-Aufruf der Alarmanlage der Aufruf einer php-Datei, die wie folgt aussieht:

```
<?php
echo "Die Alarmanlage wurde erfolgreich deaktiviert!";
echo shell_exec("sudo gpio -g mode 24 out");
echo shell_exec("sudo gpio -g write 24 0");
echo shell_exec("sudo gpio -g mode 23 out");
echo shell_exec("sudo gpio -g write 23 0");
echo shell_exec("sudo gpio -g mode 18 out");
echo shell_exec("sudo gpio -g write 18 1");
echo shell_exec("sudo mpg321 /var/www/a2.mp3");
?>
```

Die rote LED wird auf „aus“ gesetzt, die orangene LED wird auf „aus“ gesetzt und die grüne LED wird auf „ein“ gesetzt. Danach wird eine mp3-Datei ausgegeben, welche über die erfolgreiche Deaktivierung der Alarmanlage informiert. Die mp3-Dateien könnt Ihr Euch übrigens bei folgender URL nach Euren Wünschen selber erstellen und downloaden:

<http://www.fromtexttospeech.com>

Ein weiteres Beispiel, was sehr hilfreich ist, um nicht die Deaktivierung der Alarmanlage bei Betreten des Hauses zu vergessen, ist eine Begrüßungsnachricht zu Beginn der Verzögerungszeit. Hierbei wird die rote LED auf „ein“ gesetzt, die orangene LED wird auf „aus“ gesetzt und die grüne LED wird auf „ein“ gesetzt und die mp3-Datei ausgegeben. Die Alarmanlage sollte aus Sicherheitsgründen auf alle Fälle an einem anderen

versteckten Ort stehen, also nicht direkt neben dem Raspberry Pi und dem Keypad. Das sollte aber eigentlich immer der Fall sein. Die php-Datei muss dann wie folgt aussehen:

```
<?php
echo "Herzlich willkommen! Bitte als erstes die Alarmanlage deaktivieren!";
echo shell_exec("sudo gpio -g mode 24 out");
echo shell_exec("sudo gpio -g write 24 1");
echo shell_exec("sudo gpio -g mode 23 out");
echo shell_exec("sudo gpio -g write 23 0");
echo shell_exec("sudo gpio -g mode 18 out");
echo shell_exec("sudo gpio -g write 18 1");
echo shell_exec("sudo mpg321 /var/www/a1.mp3");
?>
```

Die LED's für die Fenster werden genauso gesetzt. Hier ein Beispiel ohne Sprachausgabe:

```
<?php
echo "Fenster OG offen";
echo shell_exec("sudo gpio -g mode 27 out");
echo shell_exec("sudo gpio -g write 27 1");
?>
```

Oder:

```
<?php
echo "Fenster EG zu";
echo shell_exec("sudo gpio -g mode 22 out");
echo shell_exec("sudo gpio -g write 22 0");
?>
```

Auch das Schalten der Steckdosen funktioniert so. Hier ein Beispiel ohne Sprachausgabe mit dem Hauscode der Steckdosen 11101 (siehe DIP-Schalter in der Steckdose), Steckdose 1 (Steckdose A) auf 0 setzen (ausschalten):

```
<?php
echo "11101 1 0 ";
echo shell_exec("sudo /home/pi/raspberry-remote/send 11101 1 0");
?>
```

Oder Steckdose 3 (Steckdose C) auf 1 setzen (einschalten):

```
<?php
echo "11101 3 1 ";
echo shell_exec("sudo /home/pi/raspberry-remote/send 11101 3 1");
?>
```

Die einzelnen Programschnipsel sind in jeweils eine Datei zu schreiben.

Der Dateiname kann frei vergeben werden. Die Dateiendung lautet immer *.php

Alle php-Dateien müssen auf dem Raspberry Pi in folgendes Verzeichnis kopiert werden:















/var/www/html/

Und alle mp3-Dateien in das Verzeichnis:

/var/www/

Als letztes muss nur noch die Alarmanlage konfiguriert werden.

Konfiguration der Alarmanlage:

Home Automation							
▼ Regeln							
Aktiv	✓	Nr.	Name	Bedingung	Zeitplan	Aktion	
<input checked="" type="checkbox"/>	✓	1	a1	Area 1 Verzögerung beim Betreten : Full Arm	Immer	URL: http:// /a1.php	 
<input checked="" type="checkbox"/>	✓	2	a2	Area 1 Modus : Disarm	Immer	URL: http:// /a2.php	 
<input checked="" type="checkbox"/>	✓	3	a3	Area 1 Modus : Home Arm 1	Immer	URL: http:// /a3.php	 
<input checked="" type="checkbox"/>	✓	4	a4	Area 1 Modus : Full Arm	Immer	URL: http:// /a4.php	 
<input checked="" type="checkbox"/>	✓	5	a5	Area 1 Verzögerung beim Verlassen : Full Arm	Immer	URL: http:// .a5.php	 
<input checked="" type="checkbox"/>	✓	6	f1	Türkontakt : Dachfenster oben : Offen oder Türkontakt : Dreiecksfenster : Offen	Immer	URL: http:// f1.php	 
<input checked="" type="checkbox"/>	✓	7	f2	Türkontakt : Dachfenster oben : Zu und Türkontakt : Dreiecksfenster : Zu	Immer	URL: http:// f2.php	 

In den weißen Feldern der Aktion tragt Ihr bitte die IP-Adresse Eures Raspberry Pi's ein.

Wollt Ihr auch die Anzeige der Fenster realisieren (in der Grafik z.B. für das Dachgeschoss ersichtlich), müsst ihr bei mehreren Fenstern pro Etage bei der Offen-Prüfung die Fenster-Prüfung mit „ODER“ verknüpfen und bei der Schließen-Prüfung mit „UND“ verknüpfen. Sobald ein oder mehrere Fenster offen sind leuchtet die LED.

Beim Schließen des letzten Fensters pro Etage geht die entsprechende LED aus.

Das schalten der Steckdosen ist ebenfalls über so einen Eintrag möglich, je nachdem bei welcher Bedingung dies geschehen soll.

Dann viel Spaß beim Nachbauen