Der Sonoff Schalter als Garagentorwächter.

https://www.itead.cc/sonoff-wifi-wireless-switch.html

Dies soll keine Anleitung zum Nachbau sein. Es sollte hier nur ein Projekt vorgestellt werden, in welchem ich eine Möglichkeit sehe, ein Garagentor zu einer bestimmten Zeit zufahren zu lassen, falls es noch geöffnet ist.
Gleichzeitig kann damit das Tor auch mit einem Smartphone bewegt werden.
Zusätzlich besteht die Möglichkeit, den Status des Tores anzuzeigen. - Ist das Tor auf oder zu?!?

Die Sonoff Schalter sind sehr günstige Schalter, die über Wlan angesprochen werden können.
Der originale Schalter ist für dieses Projekt absolut UNGEEIGNET, da KEIN potentialfreies Schalten des Ausganges vorgesehen ist!

Original: Eingang → N N → Ausgang
L L
Relais Schalter

Potentialfreier Umbau:

Eingang → N N → Ausgang
→ Verbindung L-N am Ausgang
L L
Relais Schalter

L-N am Eingang versorgt nur noch die Spannungsversorgung der Elektronik und des Relais.

Außerdem muss an der Platine noch eine Stiftleiste eingelötet werden, an der ein 3.3V FTDI USB to TTL Serial Adapter angeschlossen werden kann. Eine schöne Anleitung dazu gibt es auf Youtube:

https://www.youtube.com/watch?v=fN_QKOWvG1s

Nachdem EspEasy geflasht wurde und die IP des ESP bekannt ist, kann die Programmierung beginnen.

IP-in einen Browser eingeben und unter „**Tools**“ in „**Advanced**“ folgende Einstellungen vornehmen:

Welcome to ESP Easy: Garage1		
Main Config Hardware Devices Rules Tools		
Advanced Settings	**Value**	
Subscribe Template:		
Publish Template:		
Messsage Delay(ms):		
Use NTP:	x	→ Einen NTP Zeitserver eintragen.
NTP Hostname:	0.de.pool.ntp.org	
Timezone Offset:	60	→ Die Zeitverschiebung in Minuten
DST:	x	→ Daylight saving Time (im Sommer)
Syslog IP:		
Syslog Level:		
UDP port:		
Enable Serial:		→ Zugriff via (Hyper)Terminal
Serial log Level:	x	
Web log Level:	2	
Baud Rate:	2	
WD i2c Address:	115200	
Custom CSS:	0	
Use SSDP:		
Connection Failure Threshold:		
Rules:	0	→ **Rules** aktivieren!! - Sonst wird dieser Menüpunkt nicht angezeigt!
Experimental Settings	x	
I2C ClockStretchLimit:	**Value**	
Global Sync:	I2C ClockStretchLimit: 0	
	Global Sync:	
Submit		

Nun müssen die drei Ein- bzw Ausgänge konfiguriert werden – hier das Beispiel für den „button“

Welcome to ESP Easy: Garage1		
Main Config Hardware Devices Rules Tools		
Device:	Switch input	
Name:	button	← button
Delay:	0 (Optional for this device)	
IDX / Var:	1	
1st GPIO:	GPIO-0	
Pull UP:	x	
Inversed:		
Switch Type:	Switch	
Switch Button Type:	Normal	
Send Boot state:		
Send Data:		
Optional Settings Value		
Value Name 1:	state	← state

Die anderen Devices sind ähnlich anzulegen. Am Ende sollte es so aussehen wie auf der Folgenden Seite zu sehen.

- wenn man auf „Edit“ drückt, kommt man zur Koniguration des einzelnen Devices.
- Die Groß und Kleinschreibung ist wichtig. Genauso wie „state“ am Ende.
- den Namen hinter „Welcome ESP Easy“ hat man in „Config“ festgelegt.

Welcome to ESP Easy: Garage1

Main Config Hardware Devices Rules Tools

< >	Task	Device	Name	Port	IDX/Variable	GPIO	Value
Edit	1	Switch input	button		1	GPIO-0	state : 1
Edit	2	Switch input	Reed		1	GPIO-14	state : 1
Edit	3	Switch input	led		1	GPIO-13	state : 1
Edit	4	Switch input					

Jetzt können die Rules eingetragen werden:

```
on button#state do
if [button#state]=0
        gpio,13,1
        timerSet,1,1
endif
endon
```

- Wird der button am Sonoff gedrück, wird das LED (13)

- AUS geschaltet (1 bedeutet beim led – AUS)
- und der timer 1 wird für eine Sekunde gestartet

```
on Rules#Timer=1 do
  gpio,13,0
endon
```

- Wenn der Timer 1 abgelaufen ist,
- wird die LED eingeschaltet

```
on led#state do
if [led#state]=0
    gpio,12,1
    gpio,13,0
    timerSet,2,2
endif
endon
```

- Bei einer Statusänderund der „led“ überpüft ob sie
- eingeschaltet wird. Wenn ja, wird

- das Relais geschaltet (GPIO 12) → hier ist 1 = EIN!
- die LED wird sicherheitshalber noch einmal eingeschaltet
- und der Timer 2 gestartet – 2 Sekunden

```
on Rules#Timer=2 do
  gpio,12,0
  gpio,13,1
endon
```

- Nach Ablauf von 2 Sekunden
- wird das Relaies und die LED wieder ausgeschaltet (Tastenfuntion)

```
On Clock#Time=All, 19:00 do
  if [Reed#state]=0
     gpio,13,0
     timerSet,3,120
  endif
endon
```

Zur Zeitsteuerung:
- Um die hier eingegebene Zeit wird,
- Wenn der GPIO 14, der Reed Kontakt, nicht geschlossen ist, und somit das Tor offen scheint, die LED eingeschaltet.
- Das wiederum löst die Tastenfunktion von oben aus: „on led#state do“ (led entspricht GPIO-13, 0=EIN)

```
on Rules#Timer=3 do
   if [Reed#state]=0
     gpio,13,0
   endif
endon
```

- Nach 2 Minuten (120sec) wird überprüft, ob das Schließen geklappt hat und u.U. noch ein Versuch gestartet.

Danach neu starten und alles überprüfen. Die Sonoff Taste reagiert bei diesem Programm sehr langsam. Also Geduld.

Den Status der einzelnen GPIOs kann man von jedem Browser aus kontrollieren.
Hier als Beispiel die Abfrage, ob die Garage offen steht:

http://192.168.178.xxx/control?cmd=Status gpio,14

Will man die LED ausschalten, schickt man folgendes:

http://192.168.178.xxx/control?cmd=gpio,13,1

Will man die LED einschalten, schickt man folgendes:

http://192.168.178.xxx/control?cmd=gpio,13,0
- in Kombination mit den oben angeführten Rules löst das einen „Tastenimpuls“ beim Relais aus: LED geht an...

Will man somit das Garagentor mit dem Handy steuern, reicht es, die LED einmal einzuschalten. Die Rules kümmern sich um den Tastenimpuls. Schicken kann man so einen Befehl unter Android zum Beispiel mit dem App „Automate“.