

Rasenroboter Status mit ioBroker und Alexa überwachen

Da sich mein Mähroboter (Robomow MC400) manchmal beim Mähvorgang in einem Beet festfährt, habe ich nach einer Benachrichtigungsmöglichkeit gesucht.

Ich habe mich für einen Endschalter TZ-8108 in der Basisstation und einer WLAN gestützten Übertragung entschieden. Die Schalterstellung wird über einen Wemos D1 an einen MQTT-Server weitergeleitet. Im ioBroker läuft ein Javascript, das dann auf die MQTT-Nachrichten reagiert.

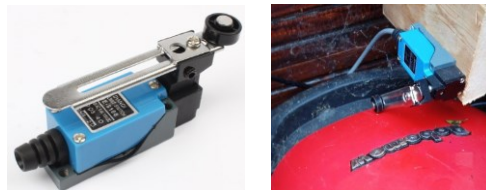


Abbildung 1 Endschalter TZ-8108

Über den ioBroker Adapter Alexa2 erfolgen entsprechende Sprachausgaben auf einen Amazon Echo Lautsprecher.

Zusätzlich wird im Vis-Adapter über ein Symbol der aktuelle Status visualisiert.

Es werden derzeit 3 Zustände signalisiert.

- Verlässt die Ladestation
- Ist in der Ladestation angekommen
- Hat sich festgefahren

Wemos D1

Firmware flashen

Als Firmware habe ich Tasmota auf den Wemos D1 geflasht.

Auf der Seite <https://github.com/arendst/Tasmota> gibt es die aktuelle Firmwaredatei (<https://github.com/arendst/Tasmota/releases>).

Für den Flaschvorgang habe ich das ebenfalls auf der Git-Seite von Tasmota angebotene Windows-Tool **Tasmotizer** (<https://github.com/tasmota/tasmotizer>) genutzt.

Nach dem Flashvorgang muss Tasmota für das eigene WLAN eingerichtet werden. Hierfür den Rechner oder das Handy mit dem vom Wemos D1 bereitgestellten WLAN (tasmota_xxxx) verbinden.

Anschließend im Browser die IP-Adresse 192.168.4.1 aufrufen und die eigene Ssid und das Passwort in die Maske eintragen und mit dem Save-Button bestätigen. Das Wemos D1 startet nun neu und verbindet sich mit dem soeben konfigurierten WLAN.

Eine ausführliche Beschreibung der Konfiguration gibt es unter der Internet-Adresse <https://tasmota.github.io/docs/Getting-Started/>

Tasmota konfigurieren

Nun im Router die IP-Adresse des Wemos D1 ermitteln und im Browser aufrufen. Die Einstellungen entsprechend der folgenden Grafiken ändern und speichern. Hierbei kann der Wemos D1 mehrfach neu starten.

Geräte-Einstellungen

Gerätetyp (Sonoff Basic)
Generic (0) ▾

D3 GPIO0 Button1	None (0) ▾
TX GPIO1 serieller Ausgang [serial out]	None (0) ▾
D4 GPIO2	None (0) ▾
RX GPIO3 serieller Eingang [serial in]	None (0) ▾
D2 GPIO4	Switch1n (82) ▾
D1 GPIO5	None (0) ▾
D6 GPIO12 Relay1	Relay1 (21) ▾
D7 GPIO13 Led1i	None (0) ▾
D5 GPIO14 Sensor	None (0) ▾
D8 GPIO15	None (0) ▾
D0 GPIO16	None (0) ▾
A0 ADC0	None (0) ▾

Speichern

Abbildung 2: Geräte-Einstellungen

MQTT-Einstellungen

Host ()
192.168.1.203

Port (1883)
1883

client (DVES_D768AB)
DVES_%06X

Benutzer (DVES_USER)
DVES_USER

Passwort ■
....

topic = %topic% (tasmota)
tasmota-02

full topic (%prefix%/ %topic%/)
Home/Garten/RobiSwitch

Speichern

Abbildung 3: MQTT-Einstellungen

Das Feld Host mit der IP des MQTT-Server füllen. Falls er einen abweichenden Port hat, muss auch dieser angepasst werden.

Sonstige Einstellungen

Vorlage

{"NAME":"Generic","GPIO":[255,255,255

Aktivieren

Passwort für Web Oberfläche ■
....

MQTT aktivieren

Name [friendly name] 1 (Tasmota)
RobiSwitch

Emulation

- keine
- Belkin WeMo Einzelnes Gerät
- Hue Bridge Mehrfachgerät

Speichern

Abbildung 4: Sonstige Einstellungen

Verkabelung des Wemos D1

Die Stromversorgung habe ich vom Stromversorgungskabel der Ladestation abgegriffen und über einen Step-Down DC-DC Wandler von 26V auf 5V reduziert. Hierzu habe ich das Anschlußkabel der Ladestation aufgetrennt und mit einer Verteilerdose erneut verbunden und die 26V mit dem Eingang des Step-Down-Wandler verbunden.



Abbildung 5: 5A DC-DC Step Down Modul XL4015

Der Endschalter wird mit dem Eingang D2 und GND des Wemos D1 verbunden.

Beide Platinen wurden in einem kleinen Gehäuse in der Nähe der Ladestation regensicher untergebracht.

ioBroker

Informationen zum Betrieb des ioBroker gibt es hier: <https://www.iobroker.net/>

Als MQTT-Broker verwende ich den Mosquitto: <https://mosquitto.org/> als eigenständigen Server. Es gibt aber für den ioBroker einen entsprechenden Adapter (MQTT Broker/Client).

Die vom Wemos D1 übertragenen MQTT Nachrichten werden von einem JavaScript im ioBroker ausgewertet und entsprechende Sprachausgaben über den Adapter Alexa2 an einen Alexa Echo Lautsprecher ausgegeben.

Zusätzlich wird der Status in einer UserVariable `'0_userdata.0.Robomow.StatusVis'` gespeichert. In Vis wird dieser Status mit dem Widget `jqui-mfd val-Door` über 3 unterschiedliche Grafiksymbole ausgegeben.

JavaScript Quellcode

Die folgenden Variablen müssen entsprechend der eigenen ioBroker Umgebung angepasst werden:

- `0_userdata.0.Robomow.StatusVis` muss manuell im ioBroker angelegt werden.
- `timeout` muss größer als die maximale Mähzeit plus Rückfahrzeit zur Ladestation sein.
- `AudioAusgabe` enthält die ID des Echo-Lautsprecher aus dem Objekt Alexa2.0.

```
// Status Rasenroboter
//(c) 2020 - Harald Woizischke
//
var switcher = 'mqtt-client.0.Home.Garten.RobiSwitch.POWER'/*Power*/; // Schalterstatus aus FHEM
var statusvis = '0_userdata.0.Robomow.StatusVis'/*StatusVis*/; // Hilfsstatus für Vis Anzeige
var counter = 0; // Zählt die Minuten wenn Roboter unterwegs
var timeout = 75; // Fehlermeldung erfolgt nach x Minuten
var timeoutrepeat = 15 // Wiederholung der Fehlermeldung nach x Minuten
var AudioAusgabe = 'alexa2.0.Echo-Devices.xxxxxxxxxxxxxxxxxx'

;
```

```

// Menütliche Abfrage des Schalters
schedule("** * * * *", function () {
  if (getState(switche).val == 'ON'){
    counter++;
    if (counter == timeout){
      counter = timeout - timeoutrepeat;
      setState(statusvis,2);
      setState(AudioAusgabe+".Commands.speak", (String('Robbi hat sich festgefahren. Er
möchte in seine Ladestation gebracht werden.')));
    }
  }
  else {
    counter = 0;
    setState(statusvis,3);
  }
});

on({id: switche, change: "any"}, function(obj) {
  if (getState(switche).val == 'ON'){
    setState(statusvis,1);
    setState(AudioAusgabe+".Commands.speak", (String('Robbi verlässt die Ladestation.')));
  }
  if (getState(switche).val == 'OFF'){
    setState(statusvis,3);
    setState(AudioAusgabe+".Commands.speak", (String('Robbi ist in der Ladestation
angekommen.')));
  }
})

```