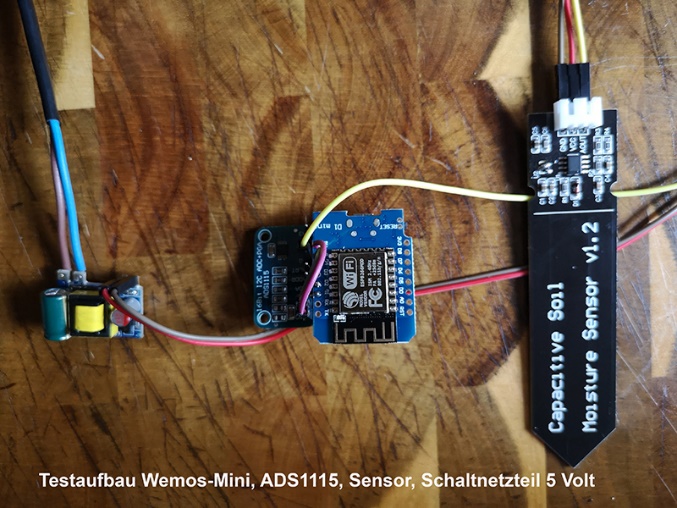
Balkonbewässerung mit Wemos Mini (NodeMCU, ESP8266) und ADS1115

* Wemos Mini
* ADS1115
* Feuchtesensor
* Schaltnetzteil



Wemos Mini entsprechend verdrahten und mit Tasmota-sensor.bin flashen .

In der Konfiguration neben den Netzwerkzugriff und MQTT ebenfalls SCL und SDA für D1 und D2 einstellen, außerdem unter Konfiguration/Logging den Wert für die Telemetry auf einen relativ kleinen Wert setzen (10).

Für diese Art des Sensors gilt, wenn bestückt wie folgt:

- TL555 und 3,3Volt Konverter -> geht direkt am Wemos Mini, funktioniert mit 3,3-5Volt

- NE555 und Konverter -> funktioniert weder mit 3,3V noch mit 5,5V (NE555 benötigt mind.4,2Volt)

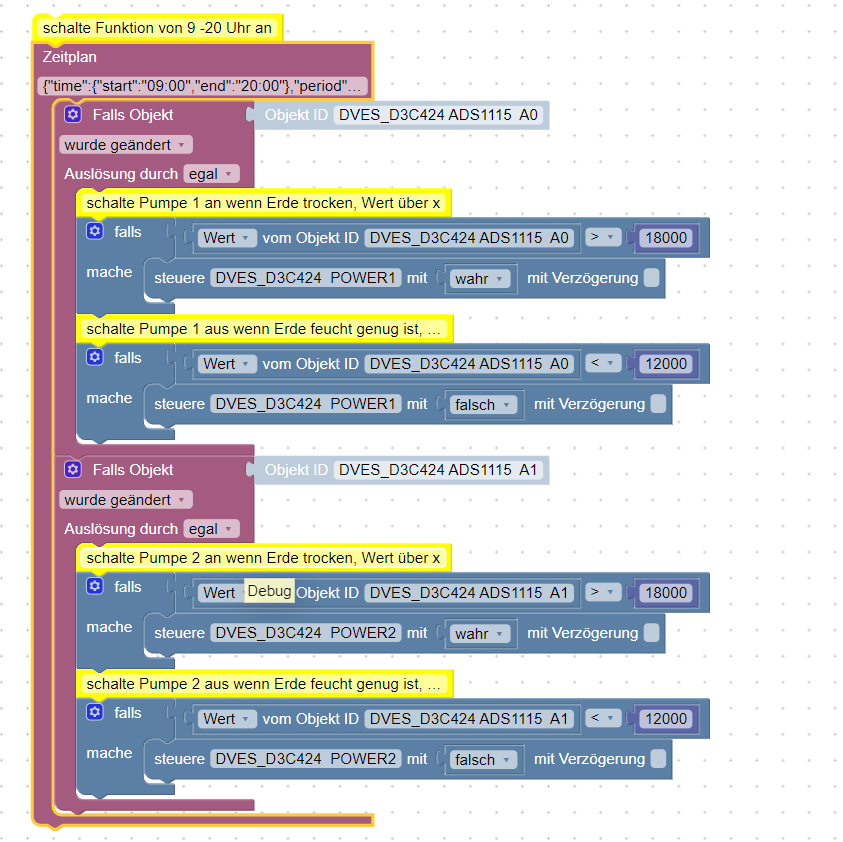
- NE555 mit Brücke für Konverter (0 ohm Widerstand) -> funktioniert nur mit 5Volt also nicht direkt am Wemos, dafür aber am ADS1115 wenn mit 5Volt betrieben.

Aus Sicherheitsgründen (Pumpen sind noch an während Blockly Timer beendet wird (20:00 Uhr)

Werden dies im Blockly expilziet nach 20 Uhr ausgeschalten. Das kann man parallel dazu auch im Tasmota Timer machen !

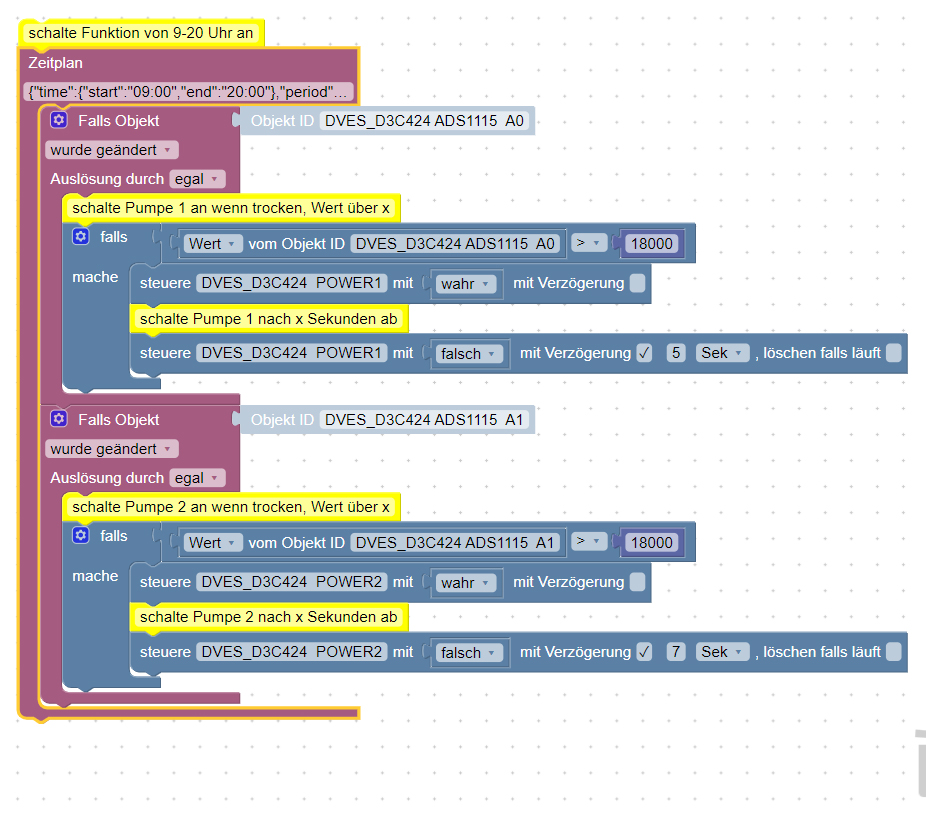
Mein Blockly Script um Tröpfler zu betreiben (Ein und Ausschaltpunkt können eingestellt werden)

\\RASPBERRY-NAS\ioBroker\Eigene Bewässerung\scripte Blockly JavaScript\Blockly\_Balkonbewässerung\_tropfer.txt

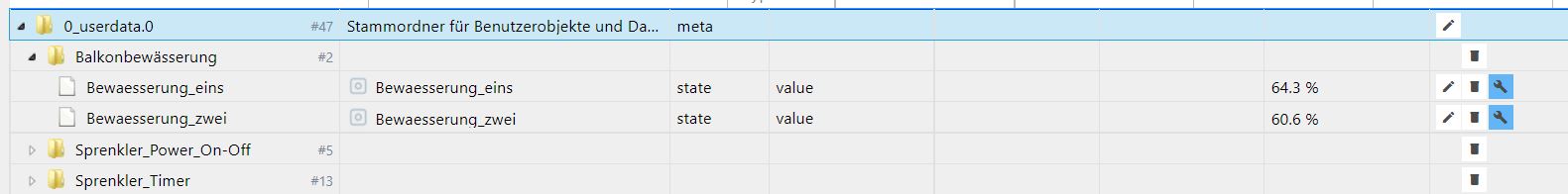


Mein Blockly Script um „zu Gießen“ (Einschaltpunkt und „Gießlänge“ können eingestellt werden)

\\RASPBERRY-NAS\ioBroker\Eigene Bewässerung\scripte Blockly JavaScript\Blockly\_Balkonbewässerung\_gießen.txt



Möchte man die erzeugten Werte die man im entsprechenden Tasmota unter den Datenpunkten findet gern als Prozentangabe haben, erstellt man „neue Datenpunkte“ unter



Die manuell erstellten Datenpunkte sollten folgende Einträge für min, max und % enthalten

"name": "Bewaesserung\_eins",

"role": "value",

"type": "number",

"desc": "Manuell erzeugt",

"unit": "%",

"min": 0,

"max": 100,

"read": true,

"write": true,

Um die Analogwerte (kleiner Wert = 100% Feucht , großer Wert = 0% Trocken) zu berechnen braucht man folgende Formel ->

Prozent = 100 - ((value - 9500) / (20400 - 9500)\*100)

Hier das JavaScript dazu ->

\\RASPBERRY-NAS\ioBroker\Eigene Bewässerung\scripte Blockly JavaScript\Sensor\_feuchte\_nach\_Prozent.zip

Um das im ioBroker unter scripte importieren zu können, macht man folgendes

Diesen Text in den Editor einfügen und als Sensor\_feuchte\_nach\_Prozent.json zu speichern.

Dann packt man das ganze als ZIP Datei ein und kann es danach direkt unter Scripte als Java Script importieren.

Bitte die eigenen Werte für Feucht = 100% und Trocken = 0% eintragen !

var Prozent, value;

on({id: "sonoff.0.DVES\_D3C424.ADS1115\_A0"/\*DVES\_D3C424 ADS1115  A0\*/, change: "any"}, async function (obj) {

  var value = obj.state.val;

  var oldValue = obj.oldState.val;

  Prozent = 100 - ((value - 9500) / (20300 - 9500)\*100);

  setState("0\_userdata.0.Balkonbewässerung.Bewaesserung\_eins"/\*Bewaesserung\_eins\*/, Math.round(Prozent\*10)/10, true);

});

on({id: "sonoff.0.DVES\_D3C424.ADS1115\_A1"/\*DVES\_D3C424 ADS1115  A1\*/, change: "any"}, async function (obj) {

  var value = obj.state.val;

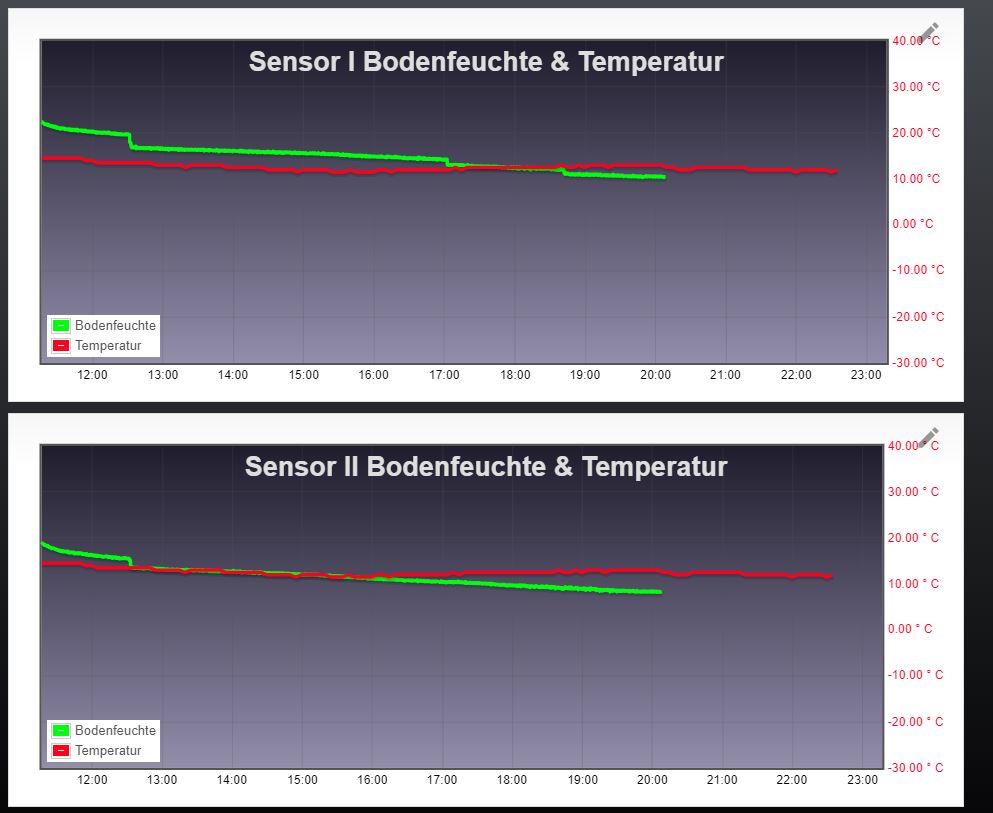
  var oldValue = obj.oldState.val;

  Prozent = 100 - ((value - 9500) / (20400 - 9500)\*100);

  setState("0\_userdata.0.Balkonbewässerung.Bewaesserung\_zwei"/\*Bewaesserung\_zwei\*/, Math.round(Prozent\*10)/10, true);

});

Natürlich kann man sich das ganze auch noch schön als Diagramm im Float erstellen



Ich hoffe es ist ggfls. Hilfreich …. Fehler sind natürlich nicht ausgeschlossen

Gruß Snupy