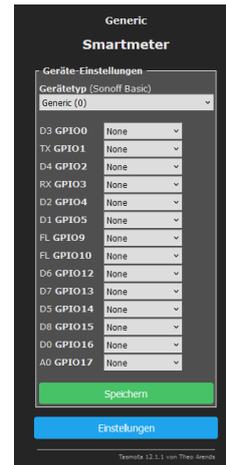


Kurzanleitung ...

... Smartmeter (EMH und Iskra MT681) mit 2x Hichi Lesekopf und 1x ESP8266 D1 auslesen und in der openWB Software im EVU-Modul einstellen ...

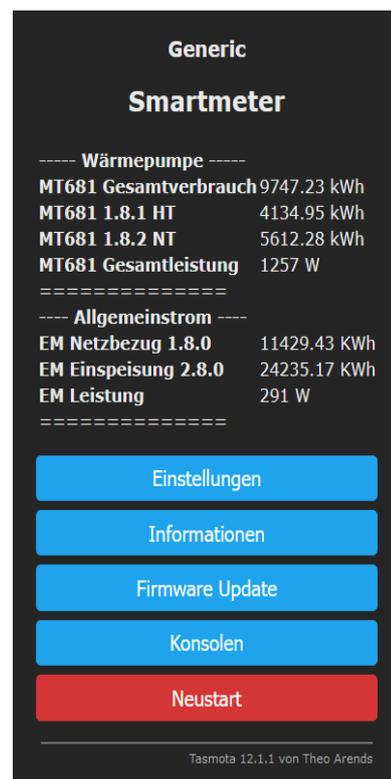
- x ESP8266 mit Script-fähiger kompilierter Tasmota Firmware flashen und verbinden des ESP mit dem heimischen WLAN ... (setze ich mal als bekannt voraus ...)
- x Unter „Geräte konfigurieren“ NIX einstellen ...
- x Die Rx der Leseköpfe an „D5“ und „Rx“ des ESP8266 löten, ebenso Vcc und GND verbinden, (Tx ist nicht zwingend notwendig ... es geht ja nur ums auslesen ...)
- x Hier der Script-Text ...
 - ... angepasst an meine Zählertypen und meine verwendeten ESP PIN's
 - ... ggf. die Script-Syntax u.v.a. nachlesen unter <https://tasmota.github.io/docs/Smart-Meter-Interface/>
 - ... Haken bei „Script aktivieren“ nicht vergessen



```
>D
>B
=>sensor53 r
>M 2

+1,3,s,0,9600,MT681
+2,14,s,0,9600,EM
1,=h ----- Wärmepumpe -----
1,77070100010800ff@1000,Gesamtverbrauch,kWh,Bezug_Gesamt,2
1,77070100010801ff@1000,1.8.1 HT,kWh,Bezug_NT,2
1,77070100010802ff@1000,1.8.2 NT,kWh,Bezug_HT,2
1,77070100100700ff@1,Gesamtleistung,W,Power_cur,0
1,=h=====
2,=h ---- Allgemeinstrom ----
2,77070100010800ff@1000,Netzbezug 1.8.0,KWh,Total_in,2
2,77070100020800ff@1000,Einspeisung 2.8.0,KWh,Total_out,2
2,77070100100700ff@1,Leistung,W,Power_curr,0
2,=h=====
#
```

- x So sieht dann die Tasmota Web UI aus ...
 - ... ggf. MQTT o.a. einstellen



x Und jetzt noch die Einbindung in die openWB Software ...

Modulkonfiguration EVU

Strombezugsmessmodul (EVU-Übergabepunkt)

Strombezugsmodul	<input type="text" value="Json"/>
Bezug URL	<input type="text" value="http://xxx.xxx.xxx.xxx/cm?cmd=status%2010"/> <small>Gültige Werte URL. Vollständige URL die die Json Antwort enthält.</small>
Json Abfrage für Watt	<input type="text" value=".StatusSNS.EM.Power_curr"/> <small>Der hier eingetragene Befehl reduziert die Json Abfrage auf das wesentliche. Im Hintergrund wird der Befehl jq benutzt. Ist die Json Antwort z.B. {"PowerInstalledPeak":4655, "PowerProduced":132, "PowerOut":897.08172362555717, "PowerSelfSupplied":234.9182763744428} so muss hier .PowerOut eingetragen werden.</small>
Json Abfrage für Bezug Wh	<input type="text" value=".StatusSNS.EM.Total_in*1000 round"/> <small>Der hier eingetragene Befehl reduziert die Json Abfrage auf das wesentliche. Im Hintergrund wird der Befehl jq benutzt. Ist die Json Antwort z.B. {"PowerInstalledPeak":4655, "PowerProduced":132, "PowerOut":897.08172362555717, "PowerSelfSupplied":234.9182763744428} so muss hier .PowerProduced eingetragen werden.</small>
Json Abfrage für Einspeisung Wh	<input type="text" value=".StatusSNS.EM.Total_out*1000 round"/> <small>Der hier eingetragene Befehl reduziert die Json Abfrage auf das wesentliche. Im Hintergrund wird der Befehl jq benutzt. Ist die Json Antwort z.B. {"PowerInstalledPeak":4655, "PowerProduced":132, "PowerOut":897.08172362555717, "PowerSelfSupplied":234.9182763744428} so muss hier .PowerSelfSupplied eingetragen werden.</small>

EVU Glättung Aus An
Kombiniert die EVU Werte der letzten x Sekunden und bildet einen Mittelwert. Sinnvoll, wenn öfter kurze Lastspitzen auftreten. Der Durchschnittswert wird auf der Hauptseite in Klammern angezeigt.

... unbedingt die Namen beibehalten ... sonst kann die Software die Wert nicht verarbeiten

Strombezugsmodul:	Json
Bezugs URL:	http://<IP-Adresse-TASMOTA>/cm?cmd=status%2010
Json Abfrage für Watt:	.StatusSNS.EM.Power_curr*1000
Json Abfrage für Bezug Wh:	.StatusSNS.EM.Total_in*1000 round
Json Abfrage für Einspeisung Wh:	.StatusSNS.EM.Total_out*1000 round

x ... fertig

(... wie beschrieben hat bei mir alles funktioniert, eine Garantie o.ä. kann ich natürlich trotzdem nicht übernehmen ... aber man kann sich sicher daran orientieren ... Stand 06.01.2023)