2019.10.11-09:00 Uhr

SML Treiber für Wasser- Gas- Strom-Zähler

Treiber erstellt von

gemu2015

auf Github veröffentlicht

www.github.com/gemu2015/Sonoff-Tasmota

Inhaltsverzeichnis:

Seite	1	Deckblatt
Seite	2	Inhalt
Seite	3	**Danke an gemu2015**
Seite	4 – 10	Beispiel Konfiguration
Seite	11 – 12	Einstellungen in Tasmota und im neuen Scripter
Seite	13 – 19	Beispiel-Script
Seite	20	Dump Modus
Seite	21	Zähler PIN
Seite	22	Besonderheiten bei Zählern
Seite	23	Erklärung eines Zählereintrages
Seite	24	Mathematik mit dem SML Treiber
Seite	25	Der METER_DESC
Seite	26	Counter und Zählerstand
Seite	27	Scripter und Zählerstände
Seite	28	Scripter und SENDMAIL
Seite	29	Stromzähler und seine Schnittstellen
Seite	30	Screenshot des Webdisplay's
Seite	31	externe Link's
Seite	32	Notizen

SML Treiber für Wasser- Gas und- Stromzähler

Einen großen Dank an „gemu2015“ für die Entwicklung des Scripter‘s und des SML Treibers

Im **„CRATION{X} SMARTHome Forum“**

wird der Beitrag unter dem Titel

„D0 Zähler SML auslesen mit Tasmota“

besprochen.

Link: https://forum.creationx.de/forum/index.php?thread/1095-d0-z%C3%A4hler-sml-auslesen-mit-tasmota/&pageNo=1

Mögliche Anpassungen im gemu2015 / Sonoff-Tasmota Fork
für Wasser, Gas u. Stromzähler
www.github.com/gemu2015/Sonoff-Tasmota

Wichtigste Frage zuerst
welches Sendungsprotokoll hat mein Stromzähler:

OBIS oder SML (Smart Message Language)?

Bitte prüfen im Datenblatt, beim Hersteller oder beim Energieversorger erfragen.

als Beispiel:

- von meierchen006
- für einen WGS_Combo Zähler (WGS = Wasser Gas Strom)

Nicht alle Anpassungen sind für jeden Notwendig, es kommt immer auf die eigene persönliche Konfiguration an.

z.B.

- welcher esp, NodeMCU oder wemosd1
- Zusatzspeicher „EEPROM AT24C256 32kb“ vorhanden
- Stromzähler obis oder SML
- Wasserzähler und wie werden die Impulse davon geliefert/gelesen
- Gaszähler und wie werden die Impulse davon geliefert/gelesen

Meine Konfiguration ist

- NodeMCU v3
- RS232 zu TTL (Max3232) für den Stromzähler

- Hager Stromzähler EHZ363W5
  - Hager optischer Kommunikationskopf (OKK) EHZ001K (für hintere Schnittstelle)

- Wasserzähler von Itron
  - Wasserzähler Abtastung „über Näherungssensor“

- Gaszähler BK G4
  - Gaszähler Abtastung „über Reedkontakt“

- EEPROM AT24C256 (32kb)

Anpassungen im gemu2015 Tasmota Fork

Stand ist der Download des Forks vom 24.09.2019 21:02 Uhr | Version 6.6.0.13

Datei:
my_user_config.h

- Zeile 39 //#define USE_CONFIG_OVERRIDE
  - #define USE_CONFIG_OVERRIDE

- Zeile 224 //#define MY_LANGUAGE de-DE
  - #define MY_LANGUAGE de-DE

- Zeile 424 #define USE_SML_M (einfügen bei serial sensors)

Datei:
user_config_override.h erstellen (aus Vorlage user_config_override:sample.h)

Ab Zeile 44 sieht meine so aus:

```
/*
#undef WIFI_IP_ADDRESS                  "0.0.0.0"
#define WIFI_IP_ADDRESS                 "192.168.1.100"
#undef WIFI_GATEWAY                     "192.168.1.1"
#define WIFI_GATEWAY                    "192.168.1.1"
#undef WIFI_SUBNETMASK                  "255.255.255.0"
#define WIFI_SUBNETMASK                 "255.255.255.0"
#undef WIFI_DNS                         "192.168.1.1"
#define WIFI_DNS                        "192.168.1.1"
*/

// -- Master parameter control --------------------
#undef  CFG_HOLDER
#define CFG_HOLDER       4626

// -- Setup your own Wifi 1 settings  ---------------
#undef  STA_SSID1
#define STA_SSID1        "Deine SSID vom WLAN"
#undef  STA_PASS1
#define STA_PASS1        "Dein WLAN Schlüssel"

// -- Setup your own Wifi 2 settings  ---------------
#undef  STA_SSID2
#define STA_SSID2        "Deine SSID 2"          //
#undef  STA_PASS2
#define STA_PASS2        " Dein WLAN Schlüssel 2"
```

```
// -- Location ----------Dein Standort----------
#undef LATITUDE
#define LATITUDE          33.372185
#undef LONGITUDE
#define LONGITUDE         5.376585

// -- Timezone -----------------------------------
#undef APP_TIMEZONE  1
#define APP_TIMEZONE  99

// -- Localization --Sprache---------------------
// If non selected the default en-GB will be used
#undef MY_LANGUAGE
#define MY_LANGUAGE  de-DE

// -- Serial sensors -----------------------------
// Add support für Smart Message Language
// (SmartMetering für Stromzähler xsns_53_SML.ino)
#undef USE_SML_M
#define USE_SML_M

// -- zusätzlicher Speicher I2C 24C256 (32KB) DEFAULT nur 4KB genutzt
#define USE_24C256

// => „nur bei sehr großem SCRIPT notwendig“
#undef EEP_SCRIPT_SIZE
#define EEP_SCRIPT_SIZE 4095        // 4KB
// #define EEP_SCRIPT_SIZE 5120     // 5KB
// #define EEP_SCRIPT_SIZE 6144     // 6KB
// #define EEP_SCRIPT_SIZE 7168     // 7KB
// #define EEP_SCRIPT_SIZE 8196     // 8KB

// -- FRIENDLY_NAME ------------------------
#undef FRIENDLY_NAME
#define FRIENDLY_NAME "WGS_Combo"

// -- WEB_Display --------------------------
#define USE_SCRIPT_WEB_DISPLAY

        // -- WEB_Display zu MQTT-------------------------
        //#define USE_SCRIPT_JSON_EXPORT
        // >J
        //,power_day:%s_d%

//--Rules oder Scripter benutzen: -----------------
//um rules zu verwenden gar nichts tun
// -- um scripter zu verwenden
#undef USE_RULES
#define USE_SCRIPT
```

```
// Veränderungen für die >D Sektion im SCRIPTER
// => „nur bei sehr großem Script notwendig“
#undef MAXVARS
#define MAXVARS 60          // Anzahl (nv) Variablen in >D im Script

#undef MAXSVARS
#define MAXSVARS 10         // Anzahl (tv) Text Variablen in >D im Script
```

/*

Kurze Erklärung zu den Parametern von oben

die Maximalwerte sind die, die du definiert hast

```
#define MAXVARS xx      => nv
#define MAXSVARS xx     => tv
```

Sollte ein Fehler im SCRIPT sein mit z.B. (zu vielen Variablen / Zeichen)
wird es in der Konsole eine Fehlermeldung geben.
Diese würde in der Konsole evtl. so aussehen:

- Script: init error -5

Dann wäre es notwendig die obigen Parameter zu ändern

Ist mit dem SCRIPT in der Konsole alles in Ordnung würde die Meldung so aussehen:

- Script: nv51, tv8, vns249, ram1234

Erklärung der Kürzel

- nv=Anzahl Zahlen Variablen (number variables)
- tv=Anzahl Text Variablen (text variables)
- vns=Anzahl Zeichen für Variablennamen (variable name size) muss unter 256 sein
- ram=von Scripter benutztes Ram
  (dazu kommt noch der Script Speicher selbst, also Default 4095 bytes)

*/

Datei:
xsns_53_sml.ino

- Wenn man Scripter benutzt ist eine Anpassung dieser Datei nicht notwendig da alles im Scripter eingestellt wird.

- Sollte aus irgendwelchen Gründen der Scripter ausfallen wird dann die Grundeinstellung dieser Datei angezeigt!

Original	Meine Anpassung
Ab Zeile 35	
// debug counter input to led for counter1 and 2 //#define DEBUG_CNT_LED1 2 //#define DEBUG_CNT_LED1 2	// debug counter input to led for counter1 and 2 #define DEBUG_CNT_LED1 12 // D6 GPIO12 blau #define DEBUG_CNT_LED2 14.// D5 GPIO 14 gelb
Ab Zeile 53	Ab Zeile 53

// max 23 chars
#if DMY_LANGUAGE==de-DE
// german web text
#define D_TPWRIN "Verbrauch"
#define D_TPWROUT "Einspeisung"
#define D_TPWRCURR "Aktueller Verbrauch"
#define D_TPWRCURR1 "Verbrauch P1"
#define D_TPWRCURR2 "Verbrauch P2"
#define D_TPWRCURR3 "Verbrauch P3"
#define D_Strom_L1 "Strom L1"
#define D_Strom_L2 "Strom L2"
#define D_Strom_L3 "Strom L3"
#define D_Spannung_L1 "Spannung L1"
#define D_Spannung_L2 "Spannung L2"
#define D_Spannung_L3 "Spannung L3"
#define D_METERNR "Zähler Nr"
#define D_METERSID "Service ID"
#define D_GasIN "Zählerstand"
#define D_H2oIN "Zählerstand"
#define D_StL1L2L3 "Ströme L1+L2+L3"
#define D_SpL1L2L3 "Spannung L1+L2+L3/3"
 	
#else
// other languages (tbd)
#undef D_TPWRIN
#undef D_TPWROUT
#undef D_TPWRCURR
#undef D_TPWRCURR1
#undef D_TPWRCURR2
#undef D_TPWRCURR3
#undef D_Strom_L1
#undef D_Strom_L2
#undef D_Strom_L3
#undef D_Spannung_L1
#undef D_Spannung_L2
#undef D_Spannung_L3
#undef D_METERNR
#undef D_METERSID
#undef D_GasIN
#undef D_H2oIN
#undef D_StL1L2L3
#undef D_SpL1L2L3
	// max 23 chars
	#if DMY_LANGUAGE==de-DE
	// german web text
	#define D_TPWRIN "Zählerstand"
	#define D_TPWROUT "Einspeisung"
	#define D_TPWRCURR "Aktueller Verbrauch"
	#define D_TPWRCURR1 "Verbrauch P1"
	#define D_TPWRCURR2 "Verbrauch P2"
	#define D_TPWRCURR3 "Verbrauch P3"
	#define D_StL1L2L3 "Ströme L1+L2+L3"
	#define D_Strom_L1 "Strom L1"
	#define D_Strom_L2 "Strom L2"
	#define D_Strom_L3 "Strom L3"
	#define D_SpL1L2L3 "Spannung L1+L2+L3/3"
	#define D_Spannung_L1 "Spannung L1"
	#define D_Spannung_L2 "Spannung L2"
	#define D_Spannung_L3 "Spannung L3"
	#define D_METERNR "Zähler Nr"
	#define D_METERSID "Hager ID"
	#define D_GASIN "Zählerstand"
	#define D_H20IN "Zählerstand"
	#define D_COUNTER "Zähler"
	
	#else
	// other languages (tbd)
	#undef D_TPWRIN
	#undef D_TPWROUT
	#undef D_TPWRCURR
	#undef D_TPWRCURR1
	#undef D_TPWRCURR2
	#undef D_TPWRCURR3
	#undef D_Strom_L1
	#undef D_Strom_L2
	#undef D_Strom_L3
	#undef D_StL1L2L3
	#undef D_Spannung_L1
	#undef D_Spannung_L2
	#undef D_Spannung_L3
	#undef D_SpL1L2L3
	#undef D_METERNR
	#undef D_METERSID
	#undef D_GASIN
	#undef D_H20IN
	#undef D_COUNTER

#define D_TPWRIN "Total-In"
#define D_TPWROUT "Total-Out"
#define D_TPWRCURR "Current-In/Out"
#define D_TPWRCURR1 "Current-In p1"
#define D_TPWRCURR2 "Current-In p2"
#define D_TPWRCURR3 "Current-In p3"
#define D_Strom_L1 "Current L1"
#define D_Strom_L2 "Current L2"
#define D_Strom_L3 "Current L3"
#define D_Spannung_L1 "Voltage L1"
#define D_Spannung_L2 "Voltage L2"
#define D_Spannung_L3 "Voltage L3"
#define D_METERNR "Meter_number"
#define D_METERSID "Service ID"
#define D_GasIN "Counter"
#define D_H2oIN "Counter"
#define D_StL1L2L3 "Current L1+L2+L3"
#define D_SpL1L2L3 "Voltage L1+L2+L3/3"
 	
#endif
 	
// JSON Strings do not translate
// max 23 char
#define DJ_TPWRIN "Total_in"
#define DJ_TPWROUT "Total_out"
#define DJ_TPWRCURR "Power_curr"
#define DJ_TPWRCURR1 "Power_p1"
#define DJ_TPWRCURR2 "Power_p2"
#define DJ_TPWRCURR3 "Power_p3"
#define DJ_CURR1 "Curr_p1"
#define DJ_CURR2 "Curr_p2"
#define DJ_CURR3 "Curr_p3"
#define DJ_VOLT1 "Volt_p1"
#define DJ_VOLT2 "Volt_p2"
#define DJ_VOLT3 "Volt_p3"
#define DJ_METERNR "Meter_number"
#define DJ_METERSID "Meter_id"
#define DJ_CSUM "Curr_summ"
#define DJ_VAVG "Volt_avg"
#define DJ_COUNTER "Count"
	#define D_TPWRIN "Total-In"
	#define D_TPWROUT "Total-Out"
	#define D_TPWRCURR1 "Current-In p1"
	#define D_TPWRCURR2 "Current-In p2"
	#define D_TPWRCURR3 "Current-In p3"
	#define D_TPWRCURR "Current-In/Out"
	#define D_Strom_L1 "Current L1"
	#define D_Strom_L2 "Current L2"
	#define D_Strom_L3 "Current L3"
	#define D_StL1L2L3 "Σ Current L1L2L3"
	#define D_Spannung_L1 "Voltage L1"
	#define D_Spannung_L2 "Voltage L2"
	#define D_Spannung_L3 "Voltage L3"
	#define D_SpL1L2L3 "ø Voltage L1L2L3"
	#define D_METERNR "Meter_number"
	#define D_METERSID "Hager ID"
	#define D_GASIN "Total-In"
	#define D_H20IN "Total-In"
	#define D_COUNTER "Count"
	
	#endif
	
	// JSON Strings do not translate
	// max 23 char
	#define DJ_TPWRIN "Total_In"
	#define DJ_TPWROUT "Total_out"
	#define DJ_TPWRCURR1 "Power_p1"
	#define DJ_TPWRCURR2 "Power_p2"
	#define DJ_TPWRCURR3 "Power_p3"
	#define DJ_TPWRCURR "Power_curr"
	#define DJ_CURR1 "Curr_p1"
	#define DJ_CURR2 "Curr_p2"
	#define DJ_CURR3 "Curr_p3"
	#define DJ_CSUM "Curr_summ"
	#define DJ_VOLT1 "Volt_p1"
	#define DJ_VOLT2 "Volt_p2"
	#define DJ_VOLT3 "Volt_p3"
	#define DJ_VAVG "Volt_avg"
	#define DJ_METERNR "Meter_number"
	#define DJ_METERSID "Hager ID"
	#define DJ_GASIN "Total-In"
	#define DJ_H20IN "Total-In"
	#define DJ_COUNTER "Count"

Ab Zeile 170	Ab Zeile 170
// select this meter #define EHZ161_1	// select this meter #define WGS_COMBO
Ab Zeile 436	Ab Zeile 436

#if METER==WGS_COMBO
#undef METERS_USED
#define METERS_USED 3
 	
struct METER_DESC const
meter_desc[METERS_USED]={
[0]={1,'c',0,10,"H20",-1,1,0}, // GPIO1
[1]={4,'c',0,50,"GAS",-1,1,0}, // GPIO4
[2]={3,'s',0,SML_BAUDRATE,"SML",-1,1,0}};
 	
const uint8_t meter[]=
 	
//---Wasserzähler--sensor53 c1----------
	#if METER==WGS_COMBO
	#undef METERS_USED
	#define METERS_USED 3
	
	struct METER_DESC const
	meter_desc[METERS_USED]={
	[0]={5,'c',0,10,"H20",-1,1,0},// GPIO5
	[1]={4,'c',0,50,"GAS",-1,1,0},// GPIO4
	[2]={3,'s',0,SML_BAUDRATE,"SML",-1,1,0}};
	
	// 3 Zähler definiert
	const uint8_t meter[]=
	
	//---Wasserzähler--sensor53 c1----------

```
//"1,=h==================|"
"1,1-0:1.8.0*255(@10000," D_H2oIN ",cbm,"
DJ_COUNTER ",4|"

//---Gaszähler-----sensor53 c2----------------------------
// bei gaszählern (countern) muss der
Vergleichsstring so aussehen wie hier
"2,=h==================|"
"2,1-0:1.8.0*255(@100," D_GasIN ",cbm,"
DJ_COUNTER ",3|"

//---Stromzähler-EHZ363W5--sensor53 d0-----------
"3,=h==================|"
//0x77,0x07,0x01,0x00,0x01,0x08,0x00,0xff
"3,77070100010800ff@1000," D_TPWRIN ",KWh,"
DJ_TPWRIN ",3|"
"3,=h==================|"
//0x77,0x07,0x01,0x00,0x10,0x07,0x00,0xff
"3,77070100100700ff@1," D_TPWRCURR ",W,"
DJ_TPWRCURR ",2|"
"3,=h ------------------------------|"
"3,=m 10+11+12 @100," D_StL1L2L3 ",A,"
DJ_CSUM ",2|"
//"3,=h ------------------------------|"
"3,=m 13+14+15/#3 @100," D_SpL1L2L3 ",V,"
DJ_VAVG ",2|"
"3,=h==================|"
//0x77,0x07,0x01,0x00,0x24,0x07,0x00,0xff
"3,77070100240700ff@1," D_TPWRCURR1 ",W,"
DJ_TPWRCURR1 ",2|"
//0x77,0x07,0x01,0x00,0x38,0x07,0x00,0xff
"3,77070100380700ff@1," D_TPWRCURR2 ",W,"
DJ_TPWRCURR2 ",2|"
//0x77,0x07,0x01,0x00,0x4c,0x07,0x00,0xff
"3,770701004c0700ff@1," D_TPWRCURR3 ",W,"
DJ_TPWRCURR3 ",2|"
"3,=h ------------------------------|"
//0x77,0x07,0x01,0x00,0x1f,0x07,0x00,0xff
"3,770701001f0700ff@100," D_Strom_L1 ",A,"
DJ_CURR1 ",2|"
//0x77,0x07,0x01,0x00,0x33,0x07,0x00,0xff
"3,77070100330700ff@100," D_Strom_L2 ",A,"
DJ_CURR2 ",2|"
//0x77,0x07,0x01,0x00,0x47,0x07,0x00,0xff
"3,77070100470700ff@100," D_Strom_L3 ",A,"
DJ_CURR3 ",2|"
"3,=h ------------------------------|"
//0x77,0x07,0x01,0x00,0x20,0x07,0x00,0xff
"3,77070100200700ff@100," D_Spannung_L1 ",V,"
DJ_VOLT1 ",2|"
//0x77,0x07,0x01,0x00,0x34,0x07,0x00,0xff
"3,77070100340700ff@100," D_Spannung_L2 ",V,"
DJ_VOLT2 ",2|"
//0x77,0x07,0x01,0x00,0x48,0x07,0x00,0xff
"3,77070100480700ff@100," D_Spannung_L3 ",V,"
DJ_VOLT3 ",2|"
"3,=h==================|"
//0x77,0x07,0x01,0x00,0x00,0x00,0x09,0xff
"3,77070100000009ff@#," D_METERSID ",,"
DJ_METERSID ",0|"
"3,=h------------------------------";
#endif
```

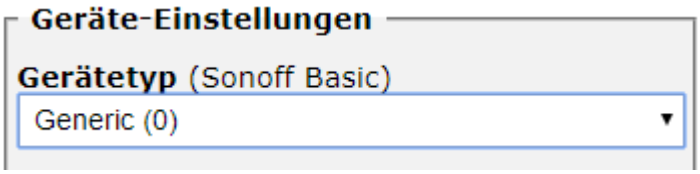
```
//"1,=h==================|"          // Zeilen
"1,1-0:1.8.0*255(@1000," D_H20IN ",cbm,"
DJ_H20IN ",3|"                       // 1

//---Gaszähler-----sensor53 c2----------------------------
// bei gaszählern (countern) muss der
Vergleichsstring so aussehen wie hier
"2,=h==================|"
"2,1-0:1.8.0*255(@100," D_GASIN ",cbm,"
DJ_GASIN ",2|"                       // 2

//---Stromzähler-EHZ363W5--sensor53-d0-----------
"3,=h==================|"
//0x77,0x07,0x01,0x00,0x01,0x08,0x00,0xff
"3,77070100010800ff@1000," D_TPWRIN ",KWh,"
DJ_TPWRIN ",3|"                      // 3
"3,=h==================|"
//0x77,0x07,0x01,0x00,0x10,0x07,0x00,0xff
"3,77070100100700ff@1," D_TPWRCURR ",W,"
DJ_TPWRCURR ",2|"                    // 4
"3,=h ------------------------------|"
//0x77,0x07,0x01,0x00,0x24,0x07,0x00,0xff
"3,77070100240700ff@1," D_TPWRCURR1 ",W,"
DJ_TPWRCURR1 ",2|"                   // 5
//0x77,0x07,0x01,0x00,0x38,0x07,0x00,0xff
"3,77070100380700ff@1," D_TPWRCURR2 ",W,"
DJ_TPWRCURR2 ",2|"                   // 6
//0x77,0x07,0x01,0x00,0x4c,0x07,0x00,0xff
"3,770701004c0700ff@1," D_TPWRCURR3 ",W,"
DJ_TPWRCURR3 ",2|"                   // 7
"3,=h==================|"
"3,=m 9+10+11 @1," D_StL1L2L3 ",A," DJ_CSUM
",2|"                                // 8
"3,=h ------------------------------|"
//0x77,0x07,0x01,0x00,0x1f,0x07,0x00,0xff
"3,770701001f0700ff@1," D_Strom_L1 ",A,"
DJ_CURR1 ",2|"                       // 9
//0x77,0x07,0x01,0x00,0x33,0x07,0x00,0xff
"3,77070100330700ff@1," D_Strom_L2 ",A,"
DJ_CURR2 ",2|"                       // 10
//0x77,0x07,0x01,0x00,0x47,0x07,0x00,0xff
"3,77070100470700ff@1," D_Strom_L3 ",A,"
DJ_CURR3 ",2|"                       // 11
"3,=h==================|"
"3,=m 13+14+15/#3 @1," D_SpL1L2L3 ",V,"
DJ_VAVG ",2|"                        // 12
"3,=h ------------------------------|"
//0x77,0x07,0x01,0x00,0x20,0x07,0x00,0xff
"3,77070100200700ff@1," D_Spannung_L1 ",V,"
DJ_VOLT1 ",2|"                       // 13
//0x77,0x07,0x01,0x00,0x34,0x07,0x00,0xff
"3,77070100340700ff@1," D_Spannung_L2 ",V,"
DJ_VOLT2 ",2|"                       // 14
//0x77,0x07,0x01,0x00,0x48,0x07,0x00,0xff
"3,77070100480700ff@1," D_Spannung_L3 ",V,"
DJ_VOLT3 ",2|"                       // 15
"3,=h==================|"
//0x77,0x07,0x01,0x00,0x00,0x00,0x09,0xff
"3,77070100000009ff@#," D_METERSID ",,"
DJ_METERSID ",0|"                    // 16
"3,=h------------------------------";  // letzte Zeile
#endif
```

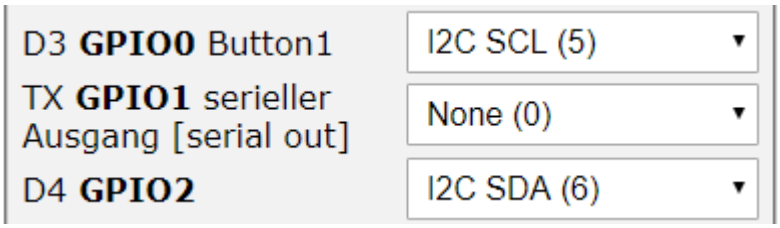
So nach diesen ganzen Änderungen kann man kompilieren

Nach dem übertragen zum NodeMCU (ESP) müssen wir in den Tasmota Einstellungen, folgendes einstellen

- 

Wenn kein zusätzlicher Speicher verwendet wird nichts einstellen!

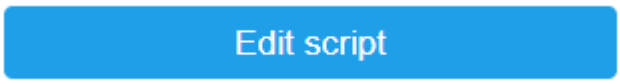
Für den zusätzlichen Speicher EEPROM AT24C256 (32kb) folgendes

- 

einstellen

- Kommt natürlich darauf an welchen (GPIO's) ihr den Speicher angeschlossen habt.

Da wir uns ja in der Datei „user_config_override.h“ entschieden haben „#define USE_SCRIPT“ zu nutzen haben wir jetzt unter Einstellungen folgenden Button:



Dort haben wir jetzt ein mächtiges Werkzeug, um unsere Tasmota Webseite des Zählers anzupassen.

So sieht das leere Fenster des Script-Editors aus, oben rechts die 4095 zeigt die Größe in Bytes für den Speicher an.

Normal ist der Speicher 1536 Bytes groß

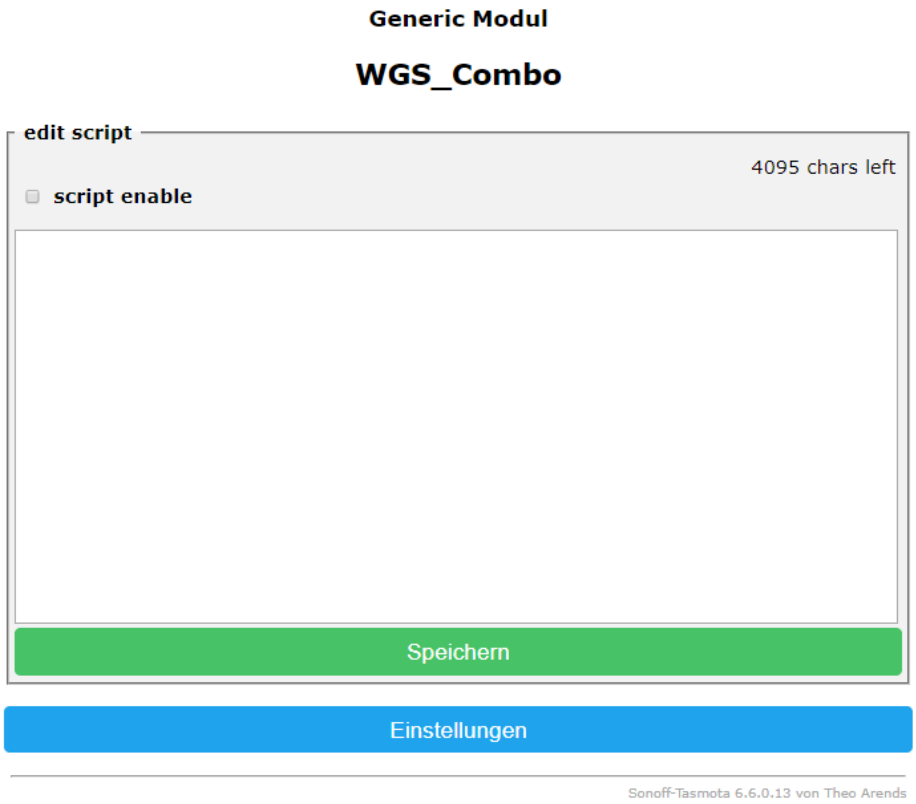
Hier ist er 4095 Bytes groß wegen dem Zusatzspeicher, den ich benutze

Man könnte den Zusatzspeicher noch größer machen,

- nach Angaben von gemu2015 höchstens 8192 Bytes
- da wir sonst Probleme mit dem RAM bekommen ab (14Kb).

Siehe auch Anpassungen in der

- „user_config_override.h“ Seite 7.



Um den Scripter einzuschalten müssen wir bei

-  einen Haken setzten.

Wie ein Script erstellt wird ist auf folgenden Links beschrieben:

- in der „scripter.md“ die sich im Download von Tasmota befindet
- auch zu finden auf der Webseite: https://github.com/gemu2015/Sonoff-Tasmota

- und auch hier gibt es Infos
  - https://forum.creationx.de/forum/index.php?thread/1791-script-editor-f%C3%BCr-tasmota/
  - https://github.com/arendst/Sonoff-Tasmota/wiki/Scripting-Language

Anbei jetzt mein Script, zu meinem WGS_Combo Zähler:

Script	Erklärungen
;yyyyyy	; Kommentare beginnen mit einem Semikolon
>D 48	>D >> Sektion D
s1="Tagesverbrauch: {m}<b><span style=" s2="aktuell: {m}<span" s3="0:00 Uhr: {m}<span" s4="Monatsanfang: {m}<span" s5="Jahresanfang: {m}<span"	String1 für Tagesverbrauch String2 für aktueller Zählerstand String3 für 0:00 Uhr Zählerstand String4 Monatsanfang Zählerstand String5Jahresanfang Zählerstand
v1=0	Variable v1 „Gas Zählerstand“ aus xsns_53_sml.ino
v2=0	Variable v2 „SML Strom Zählerstand“
v3=0	Variable v3 „Wasser Zählerstand“
v4=0	Variable v4 „Aktuelle Leistung“
v5=0	Variable v5 „Summe Ströme“
v6=0	Variable v6 „Mittelwert der Spannungen“
min=0	Variable Minuten
hr=0	Variable Stunden
wd=0	Variable Täglich 0:00 Uhr
md=0	Variable Monatsanfang 01.xx.2019 0:00 Uhr
yr=0	Variable Jahresanfang 01.01. 0:00 Uhr
farbe1="#008000"	Farbe grün
farbe2="#f00"	Farbe rot
farbe3="#1fa3ec"	Farbe hellblau
; H_2O	; Kommentar Wasser Variablen
p:wm=0	Permanente Variable Wasserzähler um 0:00 Uhr
wd=0	Variable Wasser Täglich =0
p:wma=0	Permanente Variable Wasserzähler Monatsanfang
wmn=0	Variable Wasser Monatlich =0
p:wya=0	Permanente Variable Wasserzähler Jahresanfang
wyn=0	Variable Wasser Jährlich =0
wwpr=1.79	Preis für Frischwasser 1,79 €/m³
ywpr=0	Jahres Wasserkosten =0
wapr=3.31	Preis für Abwasser 3,31 €/m³
yapr=0	Jahres Abwasserkosten =0
ywgp=65.16	Grundpreis für Wasser im Jahr
;Gas	; Kommentar Gas Variablen
p:gm=0	Permanente Variable Gaszähler um 0:00 Uhr
gd=0	Variable Gas Täglich =0
p:gma=0	Permanente Variable Gaszähler Monatsanfang
gmn=0	Variable Gas Monatlich =0
p:gya=0	Permanente Variable Gaszähler Jahresanfang

gyn=0 yhw=10.62 ggpr=0.07158 ygpr=0 yggp=107.16 ;Strom p:sm=0 sd=0 p:sma=0 smn=0 p:sya=0 syn=0 sspr=0.28421 yspr=0 ysgp=155.85 y€=0 ;CO_2 wco2=0.0000005 ywco2=0 gco2=0.000202 ygco2=0 sco2=0.000537 ysco2=0 yco2=0 yprc=10.00 y€c=0 ;---	Variable Gas Jährlich =0 „Faktor“ = Brennwert * Zustandszahl bei Gas Notwendig für Umrechnung m³ zu KWh Gaspreis =0,07158 € Jahres Gaskosten Grundpreis für Gas im Jahr ; Kommentar Strom Variablen Permanente Variable Stromzähler um 0:00 Uhr Variable Strom Täglich =0 Permanente Variable Stromzähler Monatsanfang Variable Strom Monatlich =0 Permanente Variable Stromzähler Jahresanfang Variable Strom Jährlich =0 Strompreis =0,28421 € Jahres Stromkosten Grundpreis für Strom im Jahr Jahres Gesamtkosten ; Kommentar CO_2 Variablen Wasser Emissionswert CO_2 in Gramm Wasser Jahres Emission CO_2 in t Gas Emissionswert CO_2 in Gramm Gas Jahres Emission CO_2 in t Strom Emissionswert CO_2 in Gramm Strom Jahres Emission CO_2 in t Jahres CO_2 Emission in t Preis für CO_2 Emission €/t Jahreskosten für CO_2 Emission in € ;---
>T	>T => Sektion T
v1=GAS#Total_In v2=SML#Total_In v3=H20#Total_In v4=SML#Power_curr v5=SML#Curr_summ v6=SML#Volt_avg ;---	v1= Zuweisung zu Jason Strings v2= Zuweisung zu Jason Strings v3= Zuweisung zu Jason Strings v4= Zuweisung zu Jason Strings v5= Zuweisung zu Jason Strings v6= Zuweisung zu Jason Strings ;---
>B	>B => Sektion B
->sensor53 r tper=10	Auswahl des Treibers Teleperiod=10
>S	>S => Sektion S
;Thingspeak min=int(time/5) if chg[min]>0 and v2>0 then dp3 =>Websend [184.106.153.149:80]/update? key=xxxxxxxxxxxxxxx	; Kommentar Alle 5 Minuten if chg[min]>0 and v2>0 dann dp3 senden an Thingspeak key=Schreib Key von Thingspeak

&field1=%v1% &field2=%v2% &field3=%v3% &field4=%v4% &field5=%v5% &field6=%v6% endif	Feld 1= Variable v1 Feld 2= Variable v2 Feld 3= Variable v3 Feld 4= Variable v4 Feld 5= Variable v5 Feld 6= Variable v6 endif

;Tagesverbrauch
 	
hr=hours
if chg[hr]>0
and hr==0
and v2>0
then
gm=v1
sm=v2
wm=v3
svars
endif
 	
if upsecs%tper==0{
 	
gd=v1-gm
sd=v2-sm
wd=v3-wm
}
	;Komentar Berechnung des Tagesverbauches
	
	hr=hours
	if chg[hr]>0
	and hr==0
	and v2>0
	dann
	gm=v1 => Zuweisung zu Variable v1
	sm=v2 => Zuweisung zu Variable v2
	wm=v3 => Zuweisung zu Variable v3
	svars
	endif
	
	entspricht sekunden seit Start==0
	dann
	Gas Täglich = v1-Gas 0:00 Uhr
	Strom Täglich = v2-Strom 0:00 Uhr
	Wasser Täglich = v3-Wasser 0:00 Uhr
	}

;Monatsverbrauch
 	
md=day
if chg[md]>0
and md==1
and v2>0
then
gma=v1
 	
sma=v2
 	
wma=v3
 	
svars
endif
 	
if upsecs%tper==0{
 	
gmn=v1-gma
smn=v2-sma
wmn=v3-wma
}
	;Komentar Berechnung des Monatsverbauches
	
	md=day
	if chg[md]>0
	and md==1
	and v2>0
	dann
	Gas Monatsanfang=v1 Zuweisung zu Variable v1
	Strom Monatsanfang=v2 Zuweisung zu Variable v2
	Wasser Monatsanf.=v3 Zuweisung zu Variable v3
	svars
	endif
	
	entspricht sekunden seit Start==0
	dann
	Gas monatlich = v1-Gas Monatsanfang
	Strom monatlich = v2-Strom Monatsanfang
	Wasser monatlich = v3-Wasser Monatsanfang
	}

;Jahresverbrauch yr=year if chg[yr]>0 and v2>0 then gya=v1 sya=v2	; Komentar Berechnung des Jahresverbauches yr=year if chg[yr]>0 and v2>0 dann Gas Jahresanfang =v1 Zuweisung zu Variable v1 Strom Jahresanfang =v2 Zuweisung zu Variable v2

wya=v3	Wasser Jahresanf. =v3 Zuweisung zu Variable v3
svars	svars
endif	endif
if upsecs%tper==0{	if upsecs%tper==0 then
gyn=v1-gya	Gas jährlich=v1-Gas Jahresanfang
syn=v2-sya	Strom jährlich=v2-Strom Jahresanfang
wyn=v3-wya	Wasser jährlich=v3-Wasser Jahresanfang
; Berechnung CO_2	; Berechnung CO_2
ygco2=gyn*yhw*gco2	Jahres Gas CO_2= Gas jährlich*Faktor*Gas CO_2 Emissionswert
ysco2=syn*sco2	Jahres Strom CO_2= Strom jährlich*Strom CO_2 Emissionswert
ywco2=wyn*wco2	Jahres Wasser CO_2= Wasser jährlich*Wasser CO_2 Emissionswert
;Kosten	; Berechnung Kosten
ygpr=gyn*yhw*ggpr+yggp	Jahreskosten Gas= Gas jährlich*Faktor*Gaspreis+Gasgrundpreis
yspr=syn*sspr+ysgp	Jahreskosten Strom= Strom jährlich*Strompreis+Stromgrundpreis
ywpr=wyn*wwpr+ywgp	Jahreskosten Wasser= Wasser jährlich*Wasserpreis+Wassergrundpreis
yapr=wyn*wapr	Jahreskosten Abwasser= Abwasser jährlich*Abwasserpreis
y€=ygpr+yspr+ywpr+yapr	Jahres gesamtkosten= Gaskosten+Stromkosten+ Wasserkosten+Abwasserkosten
yco2=ygco2+ysco2+ywco2	Jahres CO_2 Emissionen= Gas CO_2 +Strom CO_2 +Wasser CO_2 endif
y€c=yprc*yco2	Jahreskosten für CO_2 Emissionen
}	

>W	>W >> Sektion W (Webdispla) Das was wir als Webseite sehen
========== Wasser %s1%"color:%farbe1%">%3wd% m³ Monatsverbrauch: {m} %3wmn% m³ Jahresverbrauch: {m} %3wyn% m³ € Wasser: {m}<span style="color:%farbe2%">%2ywpr% € € Abwasser: {m}<span style="color:%farbe2%">%2yapr% € CO_2 im Jahr: {m}<span style="color:%farbe3%">%3ywco2% t ---------- Gas %s1%"color:%farbe1%">%2gd% m³ Monatsverbrauch: {m} %2gmn% m³ Jahresverbrauch: {m} %2gyn% m³ € Gas: {m}%2ygpr% € CO_2 im Jahr: {m}<span style="color:%farbe3%">%3ygco2% t ---------- Strom %s1%"color:%farbe1%">%3sd% KWh Monatsverbrauch: {m} %3smn% KWh Jahresverbrauch: {m} %3syn% KWh € Strom: {m}<span style="color:%farbe2%">%2yspr% € CO_2 im Jahr: {m}<span style="color:%farbe3%">%3ysco2% t ---------- Σ €: {m}%2y€% € ---------- Σ CO_2: {m}%3yco2% t € CO_2: {m}%2y€c% € ========== Wasser Zählerstand aktuell: {m} %3v3% m³ 0:00 Uhr: {m} %3wm% m³ Monatsanfang: {m} %3wma% m³ Jahresanfang: {m} %3wya% m³ ---------- GAS Zählerstand aktuell: {m} %2v1% m³ 0:00 Uhr: {m} %2gm% m³ Monatsanfang: {m} %2gma% m³ Jahresanfang: {m} %2gya% m³ ---------- STROM Zählerstand aktuell: {m} %3v2% KWh 0:00 Uhr: {m} %3sm% KWh Monatsanfang: {m} %3sma% KWh Jahresanfang: {m} %3sya% KWh	==========Trennlinien Wasser Tagesverbrauch: Wasser in m³ Monatsverbrauch: Wasser in m³ Jahresverbrauch: Wasser in m³ Kosten Wasser im Jahr in € Kosten Abwasser im Jahr in € CO_2 Wasser Emissionen im Jahr in t ----------Trennlinie Gas Tagesverbrauch: Gas in m³ Monatsverbrauch: Gas in m³ Jahresverbrauch: Gas in m³ Kosten Gas im Jahr in € CO_2 Gas Emissionen im Jahr in t ----------Trennlinie Strom Tagesverbrauch: Gas in KWh Monatsverbrauch: Gas in KWh Jahresverbrauch: Gas in KWh Kosten Strom im Jahr in € CO_2 Strom Emissionen im Jahr in t ----------Trennlinie Gesamt Jahreskosten ---------- Gesamt Jahres CO2 Emissionen Gesamt Jahreskosten für CO2 Emissionen ========== Wasser Zählerstände aktueller: Zählerstand 0:00 Uhr: Zählerstand Monatsanfang: Zählerstand Jahresanfang: Zählerstand ---------- GAS Zählerstände aktueller: Zählerstand 0:00 Uhr: Zählerstand Monatsanfang: Zählerstand Jahresanfang: Zählerstand ---------- STROM Zählerstände aktueller: Zählerstand 0:00 Uhr: Zählerstand Monatsanfang: Zählerstand Jahresanfang: Zählerstand

>M 3	> M >>Sektion M SML Zähler
+1,5,c,0,10,H20 +2,4,c,0,50,GAS +3,3,s,0,9600,SML //3,=h========== //3,77070100000009ff@#,Hager ID,,Meter_id,0 1,=h========== 1,1-0:1.8.0*255(@1,Zählerstand,m³,Total_In,3 2,=h========== 2,1-0:1.8.0*255(@1,Zählerstand,m³,Total_In,2 3,=h========== 3,77070100010800ff@1,Zählerstand,KWh,Total_In,3 3,=h========== 3,77070100100700ff@1,Σ Verbrauch,W,Power_curr,2 3,=h ---------- 3,77070100240700ff@1,Verbrauch L1,W,Power_p1,2 3,77070100380700ff@1,Verbrauch L2,W,Power_p2,2 3,770701004c0700ff@1,Verbrauch L3,W,Power_p3,2 3,=h========== 3,=m 9+10+11 @1,Σ Ströme,A,Curr_summ,2 3,=h ---------- 3,770701001f0700ff@1,Strom L1,A,Curr_p1,2 3,77070100330700ff@1,Strom L2,A,Curr_p2,2 3,77070100470700ff@1,Strom L3,A,Curr_p3,2 3,=h========== 3,=m 13+14+15/#3 @1,ø Spannung,V,Volt_avg,2 3,=h ---------- 3,77070100200700ff@1,Spannung L1,V,Volt_p1,2 3,77070100340700ff@1,Spannung L2,V,Volt_p2,2 3,77070100480700ff@1,Spannung L3,V,Volt_p3,2 3,=h # 	Zähler1,GPIO5;counter,Pullup=0,10ms,Name Zähler2,GPIO4;counter,Pullup=0,50ms,Name Zähler3,sml,Pullup=0,seriell 9600speed,Name Zähler3, Hager Zähler ID Zähler1. Zählerstand in m³ Zähler2. Zählerstand in m³ Zähler3. Zählerstand in KWh Zähler3, Verbrauch L1+L2+L3 in Watt Zähler3, Verbrauch L1 in Watt Zähler3, Verbrauch L2 in Watt Zähler3, Verbrauch L3 in Watt Zähler3, Ströme Zeilen 9+10+11 in Ampere Zähler3, Strom L1 in Ampere Zähler3, Strom L2 in Ampere Zähler3, Strom L3 in Ampere Zähler3, Mittelwert Spannung Zeilen 13+14+15/3 in Volt Zähler3, Spannung L1 in Volt Zähler3, Spannung L2 in Volt Zähler3, Spannung L3 in Volt #

<u>Kurze Erklärung zu den Zählern der >M Sektion:</u>

- Pullup Wiederstände sind auf den Node MCU und Wemos eingebaut diese kann man nutzen dann:
  - +1,5,c,0,10,H20 = Pullup = AUS
  - +2,4,c,0,50,GAS = Pullup = AUS
  - +3,3,s,0,9600,SML = Pullup = AUS

  - +1,5,c,1,10,H20 = Pullup = EIN
  - +2,4,c,1,50,GAS = Pullup = EIN
  - +3,3,s,1,9600,SML = Pullup = EIN

- Der Devisor (@1,) muss / sollte auch richtig gewählt werden zu seinem persönlichen Projekt!

<u>Es können bei einzelnen anderen Anwendern auch noch weitere Anpassungen Notwendig sein,</u>
<u>bestimmt ist diese Anleitung nicht vollständig!</u>

Quelle für die CO_2 Emissionswerte ist:

Für Wasser => Info der Stadt Berlin aus dem Internet 0,5 gr/m³

Für Gas und Strom => https://www.bafa.de/SharedDocs/Downloads/DE/Energie/eew_merkblatt_co2.pdf?__blob=publicationFile&v=2

Nachdem wir dies alles eingestellt haben möchten wir wissen, sendet unser Stromzähler Daten und empfangen wir diese auch wie gewünscht.

Dazu hat @gemu2015 auch einen Dump Modus eingebaut.

Wenn dieser Dump-Modus nicht gebraucht wird muss er wieder ausgeschaltet werden, sonst ist kein normaler Betrieb möglich.

Aufrufen über die Konsole:

- Sensor53 d3 zum einschalten
- Sensor53 d0 zum ausschalten

11:41:38 CMD: sensor53 d3
11:41:38 RSL: stat/sonoff/RESULT = {"Time":"2019-09-30T11:41:38","SML":{"CMD":"dump: 3"}}

So könnte der „SML Dump“ aussehen: **Start und Ende rot markiert**

19:19:49 CMD: sensor53 d3
19:19:49 RSL: stat/sonoff/RESULT = {"Time":"2019-10-01T19:19:45","SML":{"CMD":"dump: 3"}}
19:19:50 : 1b 1b 1b 1b 01 01 01 01 76
19:19:50 : 09 00 00 00 00 0f e9 f8 fa 62 01 62 00 72 63 01 01 76 01 01 09 00 00 00 00
19:19:50 : 05 4d fd a9 0b 06 48 41 47 01 0b fc f7 5e 32 01 01 63 35 a9 00 76 09 00
19:19:50 : 00 00 00 0f e9 f8 fb 62 01 62 00 72 63 07 01
19:19:50 : 77 01 0b 06 48 41 47 01 0b fc f7 5e 32 07 01 00 62 0a ff ff 72 62 01 65 05 b0 ff 12
19:19:50 : 77
19:19:50 : 77 07 81 81 c7 82 03 ff 01 01 01 01 06 49 41 45 03
19:19:51 : 77 07 01 00 00 00 09 ff 01 01 01 01 0b 26 50 41 45 01 c5 fe f7 5c 48 01
19:19:51 : 77 07 01 00 01 08 00 ff 62 82 01 62 1e 52 ff 45 01 5f 90 d7 01
19:19:51 : 77 07 01 00 01 08 01 ff 01 01 62 1e 52 ff 55 01 5a 69 c7 01
19:19:51 : 77 07 01 00 01 08 02 ff 01 01 62 1e 52 ff 53 27 10 01
19:19:51 : 77 07 01 00 10 07 00 ff 01 01 62 1f 48 00 53 01 ff 01
19:19:51 : 77 07 81 81 c7 82 05 ff 01 01 01 01 85 02 89 97 30 81 da 6f 4f 47 bf 12 00 ee ec
19:19:51 : f8 fe 62 01 62 00 72 63 02 01 71 01 63 5d e5 00 00 00 00
19:19:51 : 1b 1b 1b 1b 1a 03 66

Vorsicht Stolperfalle: beim Dump Modus

Bei mir ist der Aufruf => „sensor53 d3“

- da mein SML-Zähler an der dritten Stelle definiert ist

- entweder im Script
  - +1,5,c,0,10,H20
  - +2,4,c,0,50,GAS
  - +3,3,s,0,9600,SML

- oder auch in der xsns_53_sml.ino
  - [0]={1,'c',0,10,"H20",-1,1,0}, // GPIO1
  - [1]={4,'c',0,50,"GAS",-1,1,0}, // GPIO4
  - [2]={3,'s',0,SML_BAUDRATE,"SML",-1,1,0}};

Es ist also wichtig wo ihr euren Zähler definiert habt!

Damit Obis/SML Zähler überhaupt Daten ausgeben ist es notwendig

- eine PIN mit einer Taschenlampe an der Infrarot Diode am Zähler einzugeben.

Diese PIN bekommt man von seinem Energieversorger.

Manche Zähler geben auch dann nicht regelmäßig Daten aus, sondern nur nach einer Aufforderung.

Dazu gibt es in der xsns_53_sml.ino den Parameter:

- // #define SML_SEND_SEQ

Der Zähler (DWS74) hat einen Firmware BUG:

Dazu gibt es in der xsns_53_sml.ino den Parameter:

- // #define DWS74_BUG

Beispiel:

"1,77070100010800ff@1000," D_TPWRIN ",KWh," DJ_TPWRIN ",4|"
"3,77070100000009ff@#," D_METERSID ",," DJ_METERSID ",0|"

1, 3,	erster Eintrag = laufende Zählernummer mit Komma getrennt
77070100010800ff 77070100000009ff	danach bis @ Zeichen => Sequenz von OBIS als ASCI, oder SML als HEX ASCI
1000 #,	Skalierungsfaktor (Divisor) (kann auch negativ sein oder kleiner 0 z.B. 0.1 => Ergebnis * 10) statt des Skalierungsfaktors kann hier (nur in einer Zeile) ein # Zeichen stehen (OBIS, (SML Hager)) in diesem Fall wird ein String (keine Zahl) ausgelesen (z.B. Zähler ID) nach dem # Zeichen muss ein Abschlusszeichen angegeben werden, also bei OBIS ein) Zeichen
D_TPWRIN D_METERSID	Name des Signals in WEBUI (max 23 Zeichen)
,KWh," ,,	Einheit des Signals in WEBUI (max 7 Zeichen)
DJ_TPWRIN DJ_METERSID	Name des Signals in MQTT Nachricht (max 23 Zeichen)
,4 ,0	Anzahl der Nachkommastellen, wird hier 16 addiert wird sofort ein MQTT für diesen Wert ausgelöst, nicht erst bei teleperiod
in allen ausser der letzten Zeile muss ein | Zeichen am Ende der Zeile stehen.	
Nur am Ende der letzen Zeile steht ein Semikolon.	
max 16 Zeilen => siehe Beispiel Seite 10	

Steht in der Sequenz ein
= Zeichen am Anfang kann folgender Eintrag definiert werden:

Beispiel:

=m 9+10+11 @1," D_StL1L2L3 ",A," DJ_CSUM ",2|"

=m => mathe berechne Werte

=m 9+10+11 addiert die Ergebnisse aus den Zeilen 9,10 und 11

+ - / * werden unterstützt

das #Zeichen bezeichnet eine Konstante /#3 => geteilt durch 3

damit kann z.B. die Summe aus 3 Phasen berechnet werden
=m 13+14+15/#3 berechnet den Mittelwert aus Zeile 13+14+15#/3

=d => differenz berechnet Differenzwerte über die Zeit aus dem Ergebnis der Zeile
z.B. =d 3 10 berechnet die Differenz nach jeweils 10 Sekunden des Ergebnisses aus Zeile 3 damit kann z.B. der Momentanverbrauch aus dem Gesamtverbrauch berechnet werden, falls der Zähler das nicht direkt ausgibt

=h => html Text Zeile (max 30 Zeichen) in WEBUI einfügen,
diese Zeile zählt nicht bei Zeilenreferenzen

```
#if METER==WGS_COMBO
#undef METERS_USED
#define METERS_USED 3

struct METER_DESC const meter_desc[METERS_USED]={
  [0]={5,'c',0,10,"H20",-1,1,0},
  [1]={4,'c',0,50,"GAS",-1,1,0},
  [2]={3,'s',0,SML_BAUDRATE,"SML",-1,1,0}};

3 Zähler definiert
const uint8_t meter[]=
```

METERS_USED
muss auf die Anzahl der benutzten Zähler gesetzt werden
entsprechend viele Einträge muss der METER_DESC dann haben
(für jeden Zähler einen)

1. srcpin
   der pin für den seriellen input normalerweise 3
   => RX pin, ansonsten software serial GPIO pin
2. type
   o=obis, s=sml, c=COUNTER (z.B. Gaszähler reed Kontakt)

3. flag
   wenn
   0 dann counter ohne Pullup,
   1=mit Pullup,
   2=beutze AD Konverter ADS1115 an i2c Schnittstelle

4. params Baudrate
   bei serieller Schnittstelle,
   bei counter poll time in Millisekunden,
   wenn <0 dann Interrupt getrieben
   die negative Zahl ist Entprellzeit in Millisekunden ,
   bei ADS1115 Abtastrate

5. json prefix
   max 7 Zeichen,
   kann im Prinzip frei gesetzt werden
   dieses Prefix wird sowohl in der Web Anzeige als auch in der MQTT Nachricht vorangestellt

Nun müssen wir natürlich noch wissen wie wir einem „Counter“ seinen Zählerstand geben:

- Dem Gascounter oder Wassercounter einen Zählerstand geben für den Node MCU (esp), schließlich soll dieser ja den gleichen Zählerstand haben wie der eigentliche Gaszähler oder Wasserzähler.

- Also nehmen wir an wie oben schonmal gezeigt
  struct METER_DESC const meter_desc[METERS_USED]={
  Wasserzähler c1 [0]={5,'c',0,10,"H20",-1,1,0},
  Gaszähler c2 [1]={4,'c',0,50,"GAS",-1,1,0},
  Stromzähler SML [2]={3,'s',0,SML_BAUDRATE,"SML",-1,1,0}};

- Dann die Eingabe:
  - Wasserzähler Zählerstand 495.126 m³
    sensor53 c1 495126

  - Gaszähler Zählerstand 10128.18 m³
    sensor53 c2 1012818

  - Die Zählerstände werden
    hier ohne Punkt oder Komma eingegeben

  - Für den Obis/SML Stromzähler müssen wir nichts eingeben, der empfängt seine Daten vom Zähler.

Ganz fertig sind wir noch nicht, auch wenn wir den Scripter benutzen müssen wir dem ja auch noch einige Daten eingeben können.

Wieder mein Beispiel des WGS_Combo Zähler und meinem Script:

Einzugebende Daten als Beispiel für meinen Gaszähler

- Zählerstand aktuell
  - kommt aus dem xsns_53_sml.ino Treiber
  - oder aus dem Script

- für Zählerstand 0:00 Uhr
  - wird automatisch um 0:00 Uhr gespeichert
  - oder eingeben script >gm=12345.123;svars

- für Zählerstand Monatsanfang
  - wird automatisch am Monatsanfang gespeichert
  - oder eingeben script >gma=12345.123;svars

- für Zählerstand Jahresanfang
  - wird automatisch am Jahresanfang gespeichert
  - oder eingeben script >gya=12345.123;svars

Die hier in „rot markierten Variablen“ müssen natürlich auf die eigenen angepasst werden. da diese ja frei wählbar sind!

Das „grüne“ ist dann der Zählerstand!

Bitte beachte den (.) (punkt) statt (,) Komma

Für sendmail folgendes einstellen

am besten core pre 2.6 oder 2.42 (2.52 geht nicht)

in user_config_overwrite

folgende defines

```
#define USE_SENDMAIL
#define EMAIL_USER "user"
#define EMAIL_PASSWORD "passwd"
#define EMAIL_FROM "<mr.x@gmail.com>"
#define EMAIL_SERVER "smtp.gmail.com"
#define EMAIL_PORT 465
```

dann die mail in der Konsole testen mit z.B.

sendmail [*:*:*:*:*:*:<to@gmail.de>:test] Hallo wie gehts

im script natürlich mit => davor

man kann auch die obigen defines statt den Sternchen angeben aber so ist es kürzer

man braucht mindestens ca 20k freies RAM dafür

und durch TLS wird die binary deutlich grösser als normal

sendmail geht nur mit einigen Providern wie z.B. gmail, yahoo die den Port 465 unterstützen und kein START_TLS brauchen (das geht nicht in Arduino TLS)

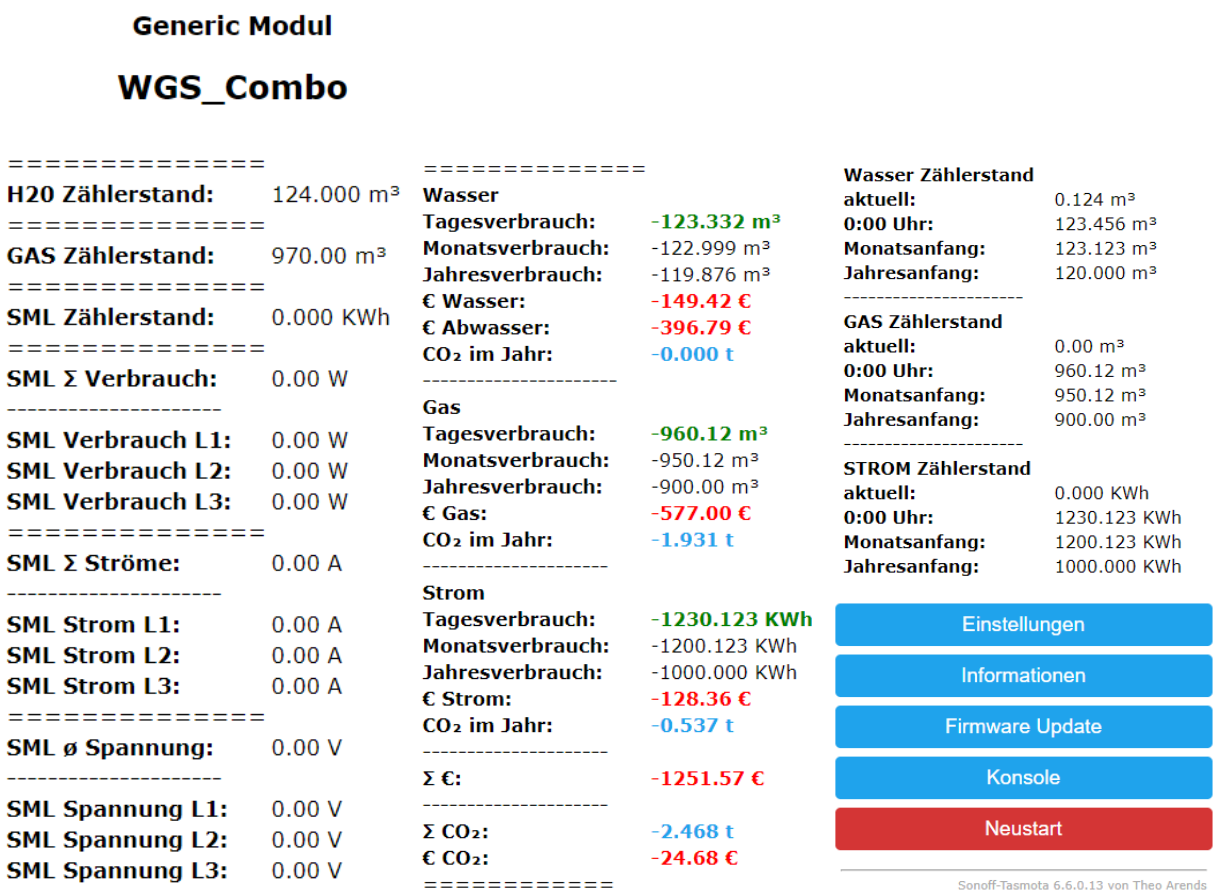
Was man noch wissen sollte:

Welche Daten von einem Strom Zähler ausgegeben werden ist abhängig vom Hersteller und der Firmware die auf dem Zähler installiert ist. Auch gibt es Unterschiede welche Daten vorderen optischen Schnittstelle oder der hinteren Schnittstelle ausgegeben werden. Es kann sogar der gleiche Zähler sein mit unterschiedlicher Firmware und dadurch auch unterschiedliche Daten ausgeben werden.

Ich nutze für mein Beispiel die hintere Schnittstelle, da dort alle Daten ausgegeben werden die für mich von Interesse sind. An der vorderen Schnittstelle gibt’s bei mir (Hager Zähler) nur 4 Daten.

Was noch zu bedenken ist,
Dass die hintere Schnittstelle nur erreichbar ist bei der Montage des Zählers vom Energieversorger und dadurch nicht selbstständig nachgerüstet werden kann!
Selbst der Elekro- Handwerksfachbetrieb kann den nicht nachrüsten da der Zähler ja verplombt ist. Natürlich lässt sich der Energieversorger dies auch bezahlen. 😊

Jetzt noch ein Screenshot von meinem Webdisplay: dreigeteilt nebeneinander



Externe Links zu besonderen Artikeln im Beitrag

- „D0 Zähler SML auslesen mit Tasmota“

- Beitrag #698 von sunburstc Platform IO installieren