

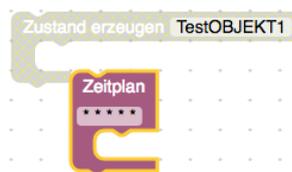
Verbrauchsauswertung aus Meßdaten der Haustechnik

Von einem Messgerät im Netz, in meinem Fall KNX und Loxone Mini Server beziehen wir einen Gesamtverbrauchswert (den Zählerstand), aus dem erzeugen wir einen Tagesverbrauch, um Mitternacht Zählerstand Vortag minus Zähler aktuell, ergibt den Tagesverbrauch. Ein Speicher history oder SQL, funktioniert bei mir leider nicht muß installiert und natürlich die Meßdaten verfügbar sein.

1. Ein neues Skript in Blockly erzeugen. Diese Ebene hilft mir als Anfänger sehr das Ganze zu verstehen.
2. Benennen des Skriptes und speichern
3. Erzeugen des/der Objekte. Dies finden wir nachher unter Objekte, das enthält dann den erzeugten Wert aus unseren Berechnungen. Den Block finden wir in System auf der linken Seite des Skriptes. Dem geben wir den Namen den nachher auch unser Wert hat.



4. 4 Objekte erzeuge ich, Verbrauch und Kosten gestern, Verbrauch und Kosten heute aktuell. Speichern und das Skript starten, dann finden wir die erzeugten Objekte unter dem Reiter Objekte unter Javascript. Der Wert ist null : ist ja noch nichts drin.
5. Da die Erstellung des Wertes ja nur um Mitternacht erzeugt werden soll, setzen wir erst einmal einen Zeitplan aus der Rubrik „Trigger“. Den Block „Zustand erzeugen“ kann man deaktivieren, oder löschen, der hat seinen „Job“ getan. Der Zeitplan wird jetzt eingerichtet, klicken auf die Fenster mit den Sternen, dann kann die Zeit eingegeben werden, bei mir 23.58. Zwei Minuten vor Mitternacht wird also diese Script ausgeführt.



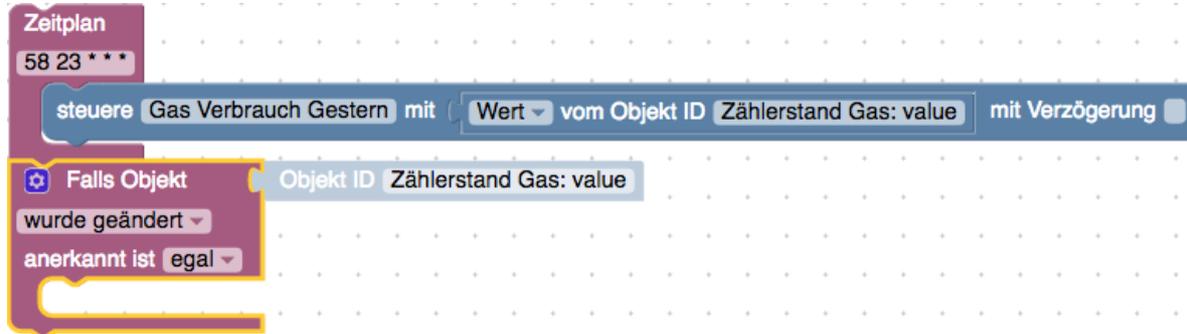
6. Nun kommt ein Block „steuere“ aus der Rubrik System. Der wird in den Zeitplan eingesetzt. Wenn man auf Objekte ID klickt, springt die Auswahl auf das Menü Objekte und man kann den zuvor erzeugten Datenpunkt auswählen den man berechnen will. Zum Beispiel „Gas Verbrauch gestern“.



7. In das weiße Puzzle Feld kommt jetzt der „Wert“ Block . Hier soll um Mitternacht der aktuelle Stand gespeichert werden. Also in der Wert ID den Zählerwert der Meßstelle auswählen (wieder unter Objekte). Die erste Steueranweisung ist fertig, um Mitternacht speichern wir den Wert in den Datenpunkt „Gas Verbrauch Gestern“



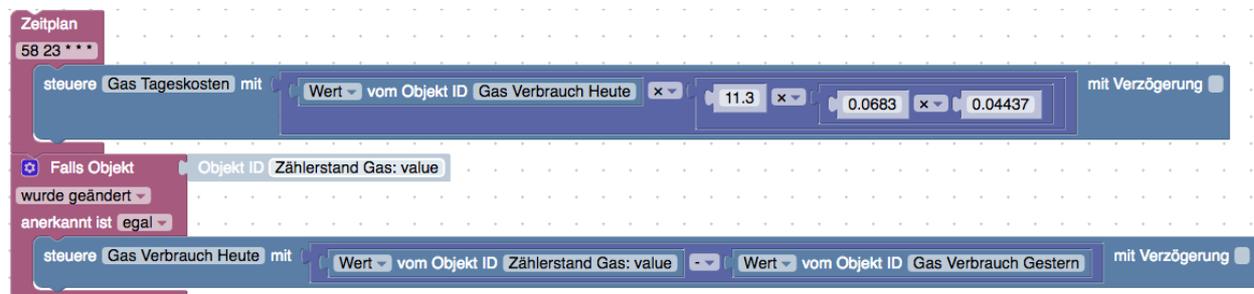
8. Um jetzt den heutigen Verbrauch berechnen zu können, muß die Bedingung „Falls Objekt geändert“ darunter setzen (NICHT darein). Es muß natürlich festgelegt werden welcher Wert zugrundegelegt wird für die Prüfung der Änderung, der in diesem Fall der aktuelle Messwert (in > Objekt ID). Wenn also am darauf folgenden Tag sich der „Gas Verbrauch Heute“ zum gestrigen geändert hat, soll er von dem gestrigen abgezogen werden.



9. Jetzt kommt ein zweiter Steuerbefehl, es soll der aktuelle Stand (um 23:58) vom dem gespeicherten Wert gestern (7.) abgezogen werden. Also ein zweiter Block „steuere“ unter den ersten. Der Wert muß berechnet werden, daher benötigt man einen Rechenblock, aus dem Menüblock Mathematik.



10. Es soll abgezogen werden, den Block mit dem „+“ kann man umstellen auf minus. Er wird wieder in das Puzzle eingesetzt. In dessen linkes Puzzle kommt der aktuelle Zählerstand, davon wird im rechten Teil der Steuerung der gestrige Wert abgezogen.



11. Wenn das soweit richtig gesetzt ist haben wir Morgen den Stand des Gaszählers um 23:58 gesichert und Übermorgen den Verbrauch des Vortages.

Zeitplan 58 23 ***

steuere Gas Tageskosten mit Wert vom Objekt ID Gas Verbrauch Heute x 11.3 x 0.0683 x 0.04437 mit Verzögerung

Falls Objekt Objekt ID Zählerstand Gas: value wurde geändert anerkannt ist egal

steuere Gas Verbrauch Heute mit Wert vom Objekt ID Zählerstand Gas: value Wert vom Objekt ID Gas Verbrauch Gestern mit Verzögerung

steuere Gas Tageskosten mit Wert vom Objekt ID Gas Verbrauch Heute x 11.3 x 0.0683 x 0.04437 mit Verzögerung

12. Jetzt möchte ich die Kosten, wieviel ich heute schon für Gas ausgegeben habe ermitteln. Dazu kommt wieder ein „steuere“ Block in den geändert Bereich, diesmal als Multiplikation des heutigen Verbrauches . Bei Gas muß m3 in kW umgerechnet werden, da der Preis pro kW angegeben ist, also $m3 \times \text{Brennwert} \times \text{Zustandszahl} \times \text{€ pro kWh}$.

13. Nun fehlen noch die Kosten fürs Gas gestern. Diese sind fix, daher müssen für über den „wurde geändert“ Block einfügen. Da die Skripte sehr schnell sind soll eine Verzögerung für die Speicherung des Tageswertes hinein, damit die Berechnung vor der Ermittlung des Zählerstandes erfolgt. Also wieder ein „steuere“ mit der Berechnung in der Verzögerung einsetzen.

javascript.0								
Stromzähler								
scriptEnabled								
Allgemeinstrom Verbrauch Gestern	Allgemeinstrom Verbrauch Gestern	state	javascript			71835	✓	⊞
Allgemeinstrom Verbrauch Heute	Allgemeinstrom Verbrauch Heute	state	javascript			100	✓	⊞
Gas Tageskosten	Gas Tageskosten	state	javascript			3.2399999999999998	✓	⊞
Gas Verbrauch Gestern	Gas Verbrauch Gestern	state	javascript			18049	✓	⊞
Gas Verbrauch Heute	Gas Verbrauch Heute	state	javascript			19	✓	⊞
Gas kosten Gestern	Gas kosten Gestern	state	javascript			3.42	✓	⊞

Zeitplan 58 23 ***

steuere Gas Tageskosten mit Wert vom Objekt ID Gas Verbrauch Heute x 11.3 x 0.0683 x 0.04437 mit Verzögerung

Ausführen timeout in 2 Sek ms

steuere Gas Verbrauch Gestern mit Wert vom Objekt ID Zählerstand Gas: value mit Verzögerung

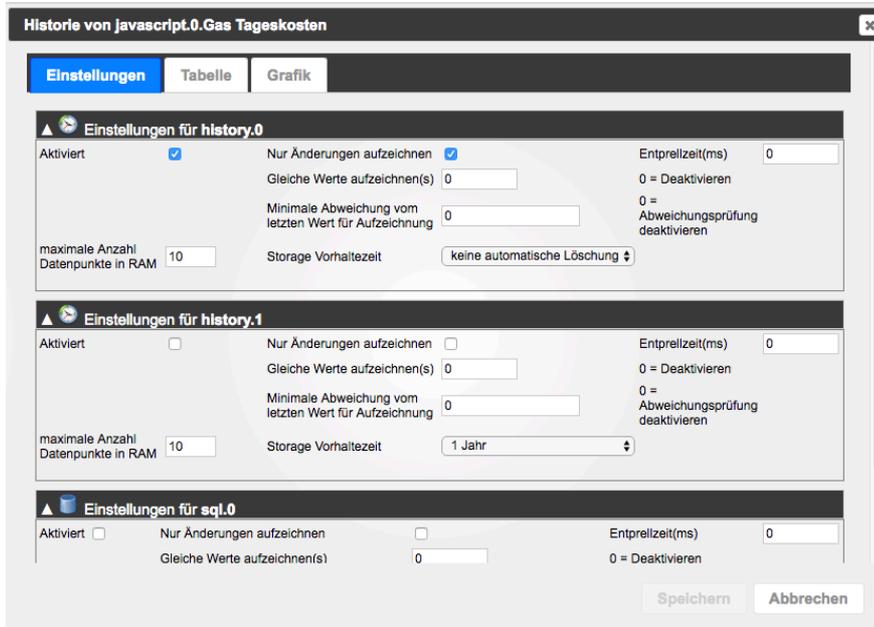
Falls Objekt Objekt ID Zählerstand Gas: value wurde geändert anerkannt ist egal

steuere Gas Verbrauch Heute mit Wert vom Objekt ID Zählerstand Gas: value Wert vom Objekt ID Gas Verbrauch Gestern mit Verzögerung

steuere Gas Tageskosten mit Wert vom Objekt ID Gas Verbrauch Heute x 11.3 x 0.0683 x 0.04437 mit Verzögerung

14. Damit werden alle vier Werte entsprechend gefüllt. Am überwachten Tag sollten die entsprechenden Zahlen zu finden sein. Ganz rechts am Zahnrad muß noch eingestellt werden, wo die Zahlen zu speichern sind, also history oder SQL.

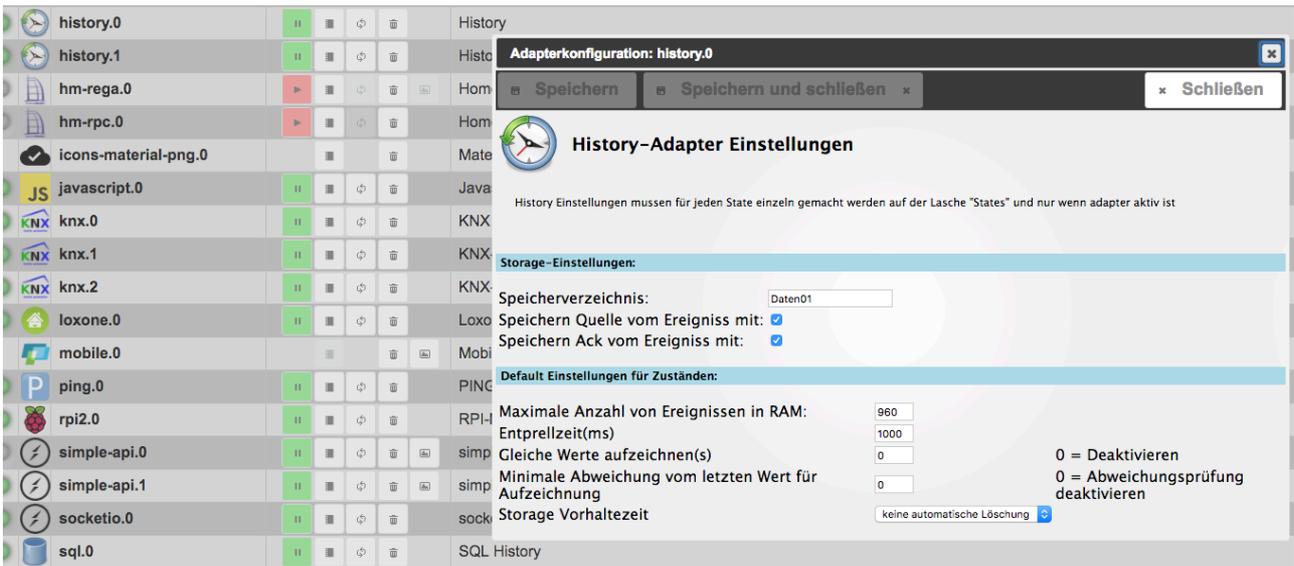
Hier das Einstellungsfenster:



In Tabelle sind dann die gespeicherten Werte zu sehen:

Wert	Bestätigt	Quelle	Zeit	Geändert
9.224257223700002	false	javascript.0	2017-12-17 12:40:39.715	
3.42	false	javascript.0	2017-12-17 12:40:39.709	
3.2399999999999998	false	javascript.0	2017-12-17 11:20:27.833	
3.2399999999999998	false	javascript.0	2017-12-17 11:20:27.833	
3.06	false	javascript.0	2017-12-17 10:35:21.220	
3.06	false	javascript.0	2017-12-17 10:35:21.210	
2.88	false	javascript.0	2017-12-17 09:55:15.337	
2.88	false	javascript.0	2017-12-17 09:55:15.335	
2.6999999999999997	false	javascript.0	2017-12-17 09:15:09.364	
2.52	false	javascript.0	2017-12-17 08:45:04.953	
2.34	false	javascript.0	2017-12-17 08:04:58.954	
2.16	false	javascript.0	2017-12-17 07:24:53.003	
1.98	false	javascript.0	2017-12-17 06:44:47.133	
1.6199999999999999	false	javascript.0	2017-12-17 06:04:41.145	
1.44	false	javascript.0	2017-12-17 04:44:29.212	
1.26	false	javascript.0	2017-12-17 04:04:23.409	
1.08	false	javascript.0	2017-12-17 03:24:17.379	
0.8999999999999999	false	javascript.0	2017-12-17 02:49:12.144	
0.72	false	javascript.0	2017-12-17 02:09:06.290	
0.54	false	javascript.0	2017-12-17 01:29:00.361	

In den Instanzen muß History noch Eingestellt werden, also der Speicherort:



Dank der großartigen, selbstlosen Hilfe von Dutchman habe ich jetzt nach nochmaligem durcharbeiten langsam verstanden, es ist auch für „Dummies“ verstehbar. Wir hatten noch einige „Debug“ Felder eingesetzt, dann erscheinen die die Werte in der Logzeile unten zur Überprüfung, das ist sehr hilfreich. Dann sah das gesamte Script zum Schluß so aus:

