Dokumentation

Betriebstundenzähler und Verbrauchsrechner

Autor: looxer01

Forum: ioBroker

Datum: 08.03.2016

geändert: 08.05.2016

Gültig ab Programmversion 0.95

Link: http://forum.iobroker.com/viewtopic.php?f=21&t=2175&sid=a2cbd8fa6237a4e049b9c6e0fc090a4b

1 Betriebsstundenzähler und Verbrauchsrechner

Die Motivation ist ein Programm zur Verfügung zu stellen, das auf die meisten Fragen von Betriebsstunden und auch Verbrauchsmessungen eine Antwort geben kann. Solche Fragen können sein: "Wie lange sind bestimmte Lampen täglich eingeschaltet" oder "Wie viel KG habe ich noch im Pellet Lager/Öllager", Wie viel Strom verbraucht mein Kühlschrank in kWH oder in Euro", wie hoch ist der bewertete Wasserverbrauch Gasverbrauch einer Periode, wie war die durchschnittliche Temperatur je Monat und wie steht mein Heizungsverbrauch in Relation dazu etc. etc.
Der Kern des Programmes ist die Messung der Zeit vom Erreichen eines Status eines Datenpunktes bis zum Verlassen des Datenpunktes.
Der einfachste Fall ist dabei das Einschalten einer Lampe. Hier wird der Status "true" gesetzt. Das Verlassen des Status wird gesetzt durch das setzen eines neuen Status. Bei einer Lampe wäre das "false". Die Zeit wird hierbei in Millisekunden gemessen und gespeichert. Darauf aufbauend findet eine Umrechnung in lesbaren Einheiten statt.
Darüber hinaus gibt es Möglichkeiten gemessene Werte zu übernehmen und umzurechnen. Hier basiert die Umrechnung nicht auf Zeit sondern auf die gemessenen Werte.
Die Darstellung der Betriebsstunden und Verbrauchswerte etc. sollte dabei in einer einzigen logischen Struktur erfolgen.

2 Funktionsweise

2.1 Grundsätzliche Funktion

Das Programm reagiert auf Veränderungen von Datenpunkten . Datenpunkte können Homematic Geräte sein oder auch selbst angelegte Datenpunkte, Systemvariablen etc. Es können bis zu 26 Datenpunkte definiert werden die entsprechend der Definition geloggt werden sollen. Wenn für bestimmte Datenpunkte Perioden definiert sind (TAG, Woche, Monat, Jahr), dann werden bei Periodenwechsel die erreichten Werte in die Periodenwerte übernommen. Wenn für bestimmte Datenpunkte die Historie aktiviert ist, dann werden die Werte auch historisch geführt, also in Monat und Jahr Variablen. Nach Wechsel des Tages (kurz nach 24 Uhr) wird überprüft der Tageswechsel durchgeführt und überprüft ob auch ein Wochen/Monats/Jahreswechsel gemacht werden muss.
Die Einstellungen werden im Wesentlichen in drei Tabellen vorgenommen. Die Gruppentabelle definiert die Hauptfunktionen. Die Log-Name Tabelle hilft bei der Namensvergabe von verschiedenen Status in Klartext und die Special Tabelle enthält Berechnungsfunktionen und Sonderfunktionen

Durch die Definition der Einstellungen in den drei Tabellen werden Datenpunkte automatisch angelegt. Diese Datenpunkte können auch wieder gelöscht werden. Siehe dazu das Kapitel: " Löschen von Datenpunkten"

2.2 Darstellung von Betriebsstunden

Die einfachste Form der Betriebsstundenerfassung erzeugt das Format ddd:hh:mm:ss. Das bedeutet das der geloggte Status ddd-Tage und hh-Stunden, mm-Minuten und ss-Sekunden eingeschaltet war. (Status kann z.B. true sein). Dies ist hilfreich für die Darstellung z.B. in VIS, da dieses Format gut lesbar ist. Es gibt aber auch viele Gründe warum man die Betriebsstunden in einem Nummer-Format dargestellt haben möchte (z.B. zur graphischen Darstellung, oder zu weiteren Berechnungen etc.). Dann kann eine einfache Umrechnung erfolgen womit z.B. eine Darstellung und Minuten, Stunden etc. möglich ist. Die Ergebnisse werden in die erzeugte Struktur geschrieben.

Anmerkung: Die 3 Tabellen zum Einstellen mögen kompliziert aussehen. Allerdings reicht z.B. für eine einfache Betriebsstundenmesser eine einzige Einstellung. Die Einstellmöglichkeiten der Tabellen stellen darüber hinaus aber viele weitere Varianten zur Verfügung, die die Möglichkeiten des Programmes erheblich ausweiten und damit auch die Generik erreicht, die gewünscht ist.

Einstellungen:

- Gruppen Tabelle - Definition der Datenpunkte
Position 1

- Special Tabelle - Umrechnungen - zu numerischen Betriebsstunden Darstellungen
Positionen 1 - 5

Beispiel der Darstellung:

2.2.1 Struktur der Darstellung

Es kam insbesondere darauf an, eine durchgängige Struktur zur Speicherung der geloggten Werte zu schaffen. Dem Nutzer sollen dabei aber Möglichkeiten gegeben werden, die Namen der Struktur Level weitestgehend selber zu bestimmen. Damit ist ein Datenchaos, wie es bei der Verwendung verschiedener Programme mit eigenen Vorgehensweisen der Datenspeicherung vorkommen kann, vorgebeugt. Bis zu sieben Level können dabei verarbeitet werden.

Einstellungen:
Gruppentabelle: 10 verschiedene Status Position 9 - 18
Gruppentabelle: Festlegung der Methode für Delta, DeltaM und CALC - Position 9
Logname: Vergebung der individuellen Namen für jeden Status getrennt

2.2.1.1 Level 1

Die Struktur ist hierarchisch und beginnt mit BSZ (Betriebsstundenzähler). Das heißt, dass alle Daten unterhalb dieses Knotens dargestellt werden. In den Einstellungen ist Name des Knotens veränderbar. Das ist z.B. hilfreich, wenn das Programm kopiert wird, z.B. zu Testzwecken und weitere Einstellungen getestet wird bevor die Einstellungen in den produktiven Knoten übernommen werden.

Einstellungen:
var sysLocation = "BSZ.System"; // Speicherort der Systemvariablen

2.2.1.2 Level 2

Im Level 2 wird dann zwischen "COUNTER" und "SYSTEM" unterschieden. Grundsätzlich befinden sich alle Nutzer-Daten unter Counter. Die System-Daten, die zu internen Berechnungen genutzt werden befinden sich unter System. Auch die Namen dieses Levels sind einstellbar. (nicht empfohlen)

Einstellungen:
var countLocation = "BSZ.Counter"; // Speicherort der Counter Variablen

2.2.1.3 Level 3

Der Level 3 kann als die thematische Beschreibung von mehreren Datenpunkten gesehen werden. So können z.B. alle Datenpunkte, die Strom messen zusammengebracht werden. Definiert wird das Thema bei den Einstellungen in der Tabelle "Gruppen" in der Position 2. Grundsätzlich gilt, dass hier frei verfahren werden kann. Zu bedenken ist jedoch, dass am Ende eine eindeutige Definition für den Daten update vorhanden sein muss. Dazu weiter unten in Level 5

Einstellungen:
Gruppentabelle: Position 2 (Feldname)

2.2.1.4 Level4

Der Level 4 kann nur bedingt durch den Nutzer beeinflusst werden. Hier werden die Methoden aufgezählt, wie z.B. TIME, DELTA, DELTAM und CALC. Die Funktionen Historie (einstellbar für jeden Datenpunkt) und Switches (Einstellbar für jeden Datenpunkt) haben hier eine Sonderstellung. Sie werden ebenfalls auf Level 4 gelistet

Einstellungen:
Gruppentabelle: Position 8 - Switch
Gruppentabelle: Position 9 - Wenn nicht TIME dann delta, deltaM, CALC
Special-Tabelle: Position 9 - AVERAGE- hier wird auf LEVEL4 der Datenpunkt AVARAGE angelegt

2.2.1.5 Level 5

Im Level 5 befinden sich grundsätzlich die Status die geloggt werden. Hier werden erstmals Werte sichtbar. Der Name des Status kann frei vergeben werden, auch wenn der Status eigentlich z.B. von Homematic Geräten stammt. (z.B. true, false). Wenn im Level 3 eine thematische Zusammenfassung eingestellt ist, dann muss -bei gleicher Methode- hier der Status auseinandergesteuert werden. Ansonsten würden Werte verschiedener Geräte auf einen Punkt zusammenlaufen (was theoretisch auch mal gewollt sein kann)

Einstellungen:
Gruppentabelle: Position 9 - 18
Logname: Vergebung der individuellen Namen für jeden Status getrennt

2.2.1.6 Level 6

Wenn in den Einstellungen (Tabelle "Gruppen") definiert worden ist, dass Perioden geführt werden sollen, dann werden geloggte Werte entsprechend in die Perioden geschrieben und bei Periodenwechsel wieder genullt. Dies kann für DAY,WEEK,MONTH und YEAR gemacht werden.

Auf Level der Perioden befindet sich der Datenpunkt "HISTORY". Von diesem Punkt gehen dann weiter die Monatlichen und Jährlichen Datenpunkte ab.

Einstellungen:
Gruppentabelle: Position 3 - History
Gruppentabelle: Position 4 - 7

2.2.1.7 Level 7+ - Vorperiode / History

Der Level 7+ dient dazu die Periodenwerte in das Feld "Vorperiode" zu schieben. Hintergrund ist, dass dies nur einmal passiert und der Wert entsprechend fluktuiert. Dies kann also für graphische Darstellungen genutzt werden oder dient einfach nur zur Information
Desweiteren lässt sich die History weiter nach Monaten bzw. Jahren aufreißen.

Einstellungen:
Gruppentabelle: Position 3 - History
Gruppentabelle: Position 4 - 7

2.3 Methoden

Methoden sind Berechnungen für Betriebsstunden und Wertedarstellung, die sich grundlegend voneinander unterscheiden. Das heißt, dass sie auch programmtechnisch unterschiedlich behandelt werden. Die Methoden TIME, Delta, DeltaM und CALC wurden implementiert und werden im folgenden erklärt.

2.3.1 TIME

Die Methode time ermittelt grundsätzlich die Zeit bei der ein bestimmter Status eingestellt wurde (Startzeit) und die Zeit bei der der Status verlassen wurde (Ende-Zeit). Die Differenzzeit ist dann Grundlage für weitere Berechnungen bzw. Darstellung. Sie stellen also die Betriebsstunden dar und werden entsprechend kumuliert. Beispiele für die Methode TIME:

- Betriebszeit für den Fernseher

- Pelletverbrauchsrechnung da Schneckenzeitabhängig (Beispiel 7,5 KG/Minute)

- Pellet-Restbestandsrechnung (was ist noch im Lager) Dabei wird der Verbrauch immer subtrahiert vom Vorbestand

- etc.

Es können bis zu 10 verschiedene Status für einen Datenpunkt geloggt werden. Da jeder Status separat in der Hierarchie dargestellt wird, kann ein eigener Name für den Status vergeben werden. Dies soll zur besseren Übersicht dienen, da z.B. ein Status "1", "2" etc. in der Darstellung nicht mehr klar wären.

Einstellungen:
Gruppentabelle: 10 verschiedene Status --> Position 9 - 18
Logname: Vergebung der individuellen Namen für jeden Status getrennt

Beispiel:

Eine Besonderheit zur Methode TIME ist der Umgang mit Geräten, die auch als LEVEL-Geräte bezeichnet werden könnten.

Kennzeichen ist, dass diese Geräte das Wort "LEVEL" in der ID tragen. Beispiele hierfür sind Dimmer oder Rollläden.

Um auch diese Geräte messbar zu machen kann in Tabelle Gruppen Position 9 ein Wert eingegeben werden. Wird dieser Wert überschritten, dann beginnt die Zeitmessung. Wird der Wert dann anschließend unterschritten endet die Zeitmessung.

So kann beispielsweise die Zeit gemessen werden an denen Rollläden geschlossen oder geöffnet sind.

Soll eine Rolllade ab 20 % als geöffnet gelten, dann muss in Position 9 "20" eingetragen werden.

Soll bei allen Werten eine Messung beginnen, dann empfiehlt sich eine "1" einzutragen.

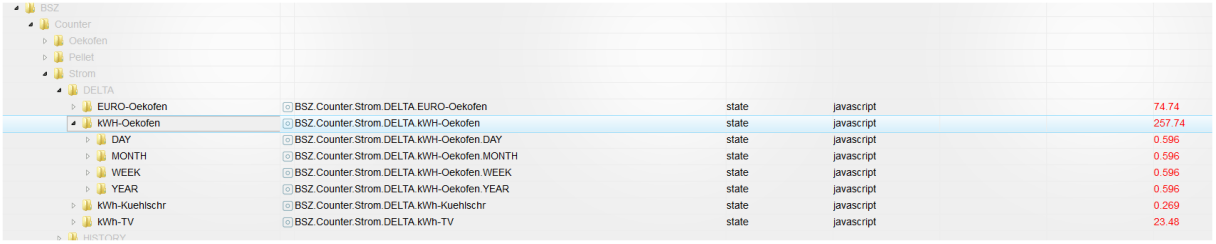
Dieser Mechanismus wird NUR aufgerufen, wenn in der ID (Tabelle Gruppen - Position 1) das Wort LEVEL enthalten ist.

2.3.2 DELTA

Delta ist für die Stromverbrauchsmessung implementiert worden. Dabei geht wird ein Wert gemeldet (z.B. vom HM Energiemesser oder auch über modbus oder S7). Der Wert stellt beispielsweise den aktuellen Zählerstand dar. Der aktuelle Zählerstand wird vom letzten Zählerstand abgezogen. Damit ergibt sich das Delta was gleich dem Verbrauch ist. Über weitere Umrechnungen kann der Verbrauch in kWh oder Euro angezeigt bzw. gespeichert werden. Eine Besonderheit ist das Rücksetzen des Zählers was u.U. zu einem unrealistisch hohen Verbrauch positiv oder negativ führen könnte. Dazu ist die Logik der Varianzen eingeführt - siehe dazu "Umgang mit Delta Varianzen"

Einstellungen:
Gruppentabelle: Position 9 für die Definition als Delta
Special-Tabelle: Position 1 - 5 für die Berechnungen

Beispiel:



2.3.3 DELTAM

DeltaM (delta-Minus) ist implementiert worden , um z.B. Sensoren, die eine Füllstandsmessung vornehmen zu unterstützen. Im Grunde ist die Rechnung ähnlich wie bei Delta nur mit umgekehrten Vorzeichen, da der Tankbestand nicht zunimmt sondern stetig abnimmt. Füllstandssensoren messen nicht 100% exakt womit sich immer eine Abweichung ergibt, die sich bei der nächsten Messung möglicherweise ausgleicht. Auch müssen Tankvorgänge berücksichtigt werden und dürfen nicht als negativer Verbrauch interpretiert werden. - siehe dazu "Umgang mit Delta Varianzen"

Einstellungen:
Gruppentabelle: Position 9 - für die Definition als DeltaM
Special-Tabelle: Position 1 - 5 für Berechnungen
Special-Tabelle: Position 7 für Deltavarianzen

2.3.4 CALC

CALC ist von allen Methoden die einfachste. Sie übernimmt lediglich einen gemeldeten Wert und rechnet ihn noch um entsprechend der Einstellungen. Eine Kumulation oder Subtraktion findet aber nicht statt.

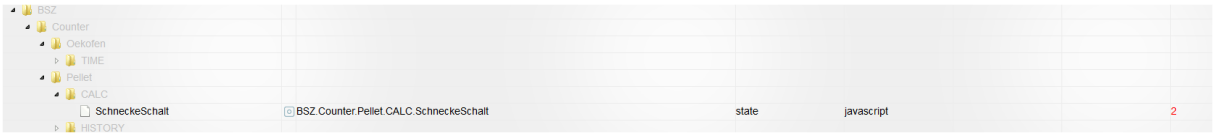
Beispiele für CALC sind

- Workaround für boolesche Werte. Da diese Werte ja im System als true und false dargestellt werden können diese nicht in Graphen gezeigt werden. Eine Umrechnung durch Addition von 1 z.B. erzeugt Werte für eine Lampe von 1 = an und 2 = aus. Wenn die Addition zudem variiert (z.B. +2 für Lampe2 und +3 für Lampe3) für verschiedene Verbrauche können diese in einem einzigen Graphen dargestellt werden.

- Temperaturen können 1:1 übernommen. Die Temperaturen können dann auch als Durchnittswerte für z.B. DAY, WEEK, MONTH, YEAR berechnet werden(Durchschnittstemperaturen) . Siehe auch die Dokumentation zur Funktion der Durchschnittsberechnung.

Einstellungen:
Gruppentabelle: Position 9 - für die Definition als CALC
Special-Tabelle: Position 1 - 5 für Berechnungen

Beispiel:



2.4 Funktionen

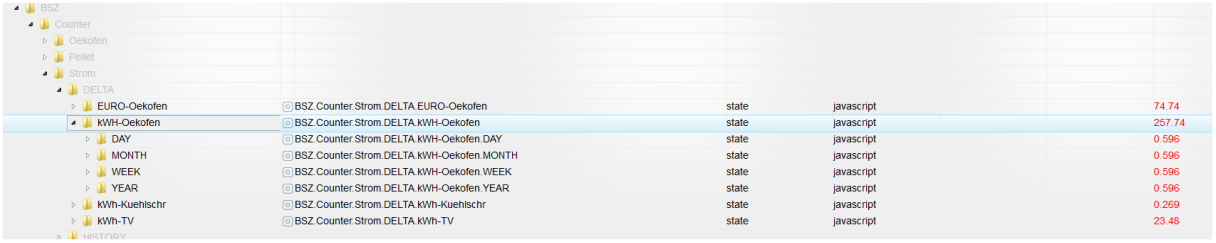
Funktionen sind Features, die zusätzlich zu den Methoden entwickelt worden sind, die Methoden also ergänzen.

2.4.1 Perioden

Jede Periode kann einzeln aktiviert werden, also für Tag, Woche, Monat und Jahr. Dabei wird automatisch in Level 7 die Vorperiode mit angelegt. Perioden machen nicht in allen Fällen Sinn (wahrscheinlich aber in den meisten Fällen) und sollten nur dann angelegt werden, wenn wirklich Interesse an den Periodenwerten besteht. Dies kann z.B. der Fall sein, wenn auch graphisch Ergebnisse dargestellt werden sollen oder auch einfach um wichtige Werte und deren Entwicklung zu verfolgen. Perioden werden zurückgesetzt bei Periodenwechsel. Die zurückgesetzten Werte werden dann in die Vorperiode geschrieben. Nach einem erneuten Periodenwechsel sind auch die Werte dann wieder überschrieben.

Einstellungen:
Gruppentabelle: Position 4 - 7 - für die Definition jeder Periode

Beispiel:



2.4.2 Historie

Wenn auch längerfristig ein Interesse besteht bestimmte Werte zu verfolgen, dann bietet sich die Funktion Historie an. Historie kann nur in Monaten oder Jahren erfolgen. (aufgrund des Datenaufkommens wurde verzichtet für Wochen und Tage Historie anzulegen). Historie macht Sinn für Werte wie die Verfolgung des Ölverbrauches über längere Zeit - also auch über Jahre, um Einsparmaßnahmen zu verfolgen oder auch den Einfluss der Jahreszeiten zu sehen.

Einstellungen:
Gruppentabelle: Position 3

Beispiel:

2.4.3 Schließen von Perioden

Kurz nach Mitternacht sollten die Perioden genullt werden. Dabei wird festgestellt ob es sich um einen Tagesabschluss, einen Wochen-Monats oder Jahresabschluss handelt. Die Periodenwerte werden dabei genullt und in die Vor-Periode geschoben. Wenn Historie eingeschaltet ist, dann werden die Werte zusätzlich in die Historienwerte geschrieben

Einstellungen- Beispiel hier jeden Tag um 00:07
var TimeSetStunde = "00"
var TimeSetMinute = "07"

2.4.4 Switches

Switches ist lediglich ein Zählwerk. Für die Methode TIME wird dabei immer gezählt wenn ein kompletter Statuswechsel erfolgt ist als z.B. AN und AUS. Dann wird eins addiert, also gezählt.

Für die anderen Methoden wird immer 1 addiert, wenn die Bedingung zutrifft. Übersetzt bedeutet dies z.B. für den Strommesser, dass für jede Messung 1 addiert wird. Switches sollten nur aktiviert werden, wenn wirklich Interesse besteht die Messvorgänge zu protokollieren. (Anm.: Switches werden später genutzt für die noch zu entwickelnde Funktion zur Berechnung von Durchschnittswerten)

Einstellungen:
Gruppentabelle: Position 8

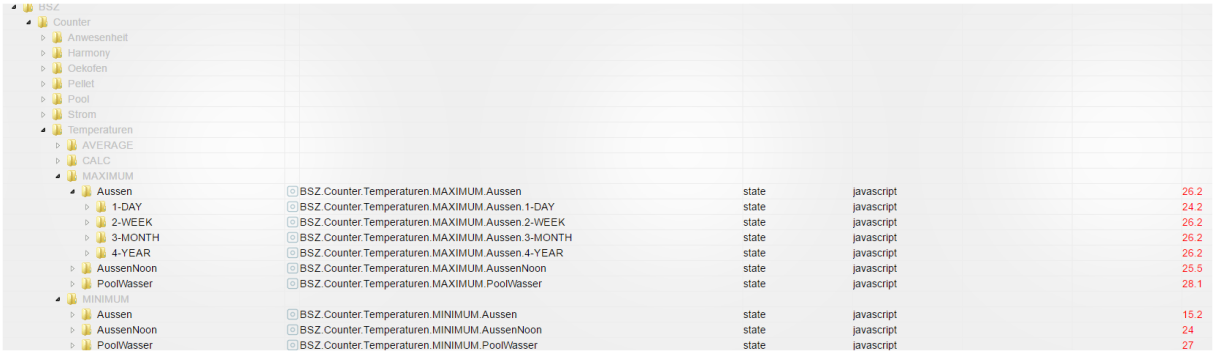
Beispiel:

2.4.5 Minimum / Maximum

Die Minimum Maximum Funktion erlaubt es die höchsten bzw. niedrigsten gemessenen Werte zu dokumentieren. Falls Perioden gewählt sind, dann werden diese Werte auch in Minimum/Maximum Wert für den letzten Tag, letzte Woche, letzter Monat und letztes Jahr festgehalten

Die Funktion kann für alle Methoden gewählt werden. Sinnvoll ist dies z.B. für Temperaturen (Methode CALC), um festzustellen wie hoch oder niedrig die jemals gemessene Temperatur war oder z.B. wie maximal kalt es in der letzten Nacht war.

Die Aktivierung erfolgt über die Tabelle "special" durch eintragung von "true" in Spalte 12 "Min/Max"



2.4.6 Umrechnungen

Alle Werte können umgerechnet werden. Die Umrechnungen sind also eine zentrale Funktion und können in Tabelle Special definiert werden. Die Reihenfolge der Berechnung ist immer : ((Wert + Addition1) * Faktor / Divisor) + Addition2

- Rundung: hier kann eingestellt werden auf wie viele Stellen gerundet werden soll. Die Rundung wird nur angewendet für die Darstellung im "COUNTER" Bereich also für den Nutzer. In den "SYSTEM" Bereich wird immer unbegründet geschrieben. Ansonsten würden sich Rundungsfehler addieren zu beträchtlichen Abweichungen. Es empfiehlt sich z.B. für kWh eine 3 einzutragen, für KG und EURO eine 2

- Addition1: hier wird der angegebene Wert addiert/subtrahiert zum Wert, der von der Methode ermittelt wurde. Bei TIME ist dies die Zeitdifferenz in Millisekunden, bei Delta ist dies der Delta-Wert zwischen letzten und aktuellen Wert. Addition1 kann genutzt werden, um z.B. den Wert 1,2,3 etc. zu booleschen Werten zu addieren. (siehe auch Methode CALC)

- Faktor ist ein Multiplikator. Es können auch negative Werte eingegeben werden. Ein Faktor kann z.B. der Preis je kWh Stunde sein.

- Divisor
Im Grunde lässt sich alles durch den Faktor darstellen. Allerdings können die Nachkommastellen für den Faktor dann recht groß werden. Der Divisor erleichtet also die Darstellung. Für die Umrechnung für die Methode TIME in Sekunden sollte also hier 1000 eingegeben werden, für Minuten 60000 und für Stunden 3.600.000

- Addition2 verhält sich wie addition1 wird aber erst addiert, wenn die vorherigen Berechnungen abgeschlossen sind.

Einstellungen:
Special-Tabelle: Position 1 - 5 für Berechnungen

2.4.7 Logikerweiterungen

Es ist die Möglichkeit implementiert worden eigene Logik (Programm-Code) zu verwenden. Ein Beispiel ist ebenfalls mit implementiert für Dekoren Heizungen, die über Modbus angeschlossen sind und über ein Pelletsilo mit integrierter Schnecke verfügen. Die o.g. Berechnungslogik wird auch verwendet, wenn eine Logikerweiterung geschrieben wurde. Die eigene Logik wird aber zuerst ausgeführt, d.h., dass der Wert aus der Erweiterung kommt

Einstellungen:
Special-Tabelle: Position 6. - Die hier eingegebene Logik muss dann implementiert werden

Name der Funktion: function individual(funktion, nummer, runtime)
Die Übergabewerte zur Funktion sind:

- funktion = die Funktion, die in Tabelle "Special" eingeben wird
- nummer = die aktuelle bearbeitete Nummer aus der Tabelle "Gruppe"
- runtime = Wert zur weiteren Verarbeitung z.B. Millisekunden. Der Wert kommt aus dem Programm und wird dann entsprechend verändert an das Programm zurückgegeben. Es wird immer ein Rohwert an die Funktion gegeben. Die Berechnungsfunktionen aus der Tabelle "Specials" werden erst anschließend angewendet

Die Funktion wird wie unten gezeigt eingebaut. Dabei muss der Name in Großbuchstaben abgefragt werden, auch wenn der Name in Tabelle "Specials" in Kleinbuchstaben definiert ist. Die Individualtfunktionen befinden sich am Ende des Skriptes

```
 if(funktion === "OEKOFEN") {

....Logik

return runtime;

}
```

2.4.8 Umgang mit Delta Varianzen

Delta-Varianzen können auftreten in den Methoden Delta und DeltaM. Die Gründe für Varianzen können unterschiedlich sein

- Das messende Geräte könnte nicht 100 % messen. Damit würden mal plus und minus-Varianzen auftreten.

- Bei einer Tankmessung müssen auch Tankfüllungen betrachtet werden.

- Ein Energiemesser - insbesondere die von HM- kann von Zeit zu Zeit zurückgesetzt werden

Wenn nichts in der Special Tabelle eingegeben wird, dann wird ein absoluter Wert von 100 angenommen. 100 bedeutet z.B. bei der Methode Delta und bei der Stromverbrauchsmessung "Watt", bei einer Messung eines Öltanks kann es Liter bedeuten.
Dabei werden alle Messungen die eine Differenz von 100 nicht überschreiten als positiver bzw. negativer Verbrauch berechnet. sollte der Wert überschritten werden, dann wird kein Verbrauch angenommen. Der Zählerstand bzw. Tankstand wird jedoch korrigiert.

Einstellungen:
Special-Tabelle: Position 7. - Wenn nichts eingegeben wird, dann wird 100 angenommen. (Default Wert)

2.4.9 Quittierung (ack)

Homematic Geräte senden üblicherweise eine Quittierung zu ioBroker um sicherzugehen, dass die Schaltung tatsächlich erfolgte. Der BSZ reagiert normalerweise erst, wenn diese Quittierung erfolgte. Es gibt aber Gründe warum darauf nicht gewartet werden soll oder es erfolgt gar keine Bestätigung (ack), da möglicherweise der verwendete Datenpunkt gar kein HM-Gerät ist, sondern ein reiner ioBroker-Datenpunkt.
Für diesen Fall kann definiert werden, dass der Trigger bei ack = false erfolgen soll.

Einstellungen:
Special-Tabelle: Position 8. - Wenn nichts eingegeben wird (''), dann wird true angenommen (Default Wert). True heißt warten auf ack = true, false heißt bei false wird der Trigger ausgelöst.

2.4.10 Durchnittsfunktion

Die Durchnittsfunktion wird in Tabelle "special" Position 9 durch setzen von "true" aktiviert. Diese Funktion macht besonders für die Methoden TIME und CALC Sinn. Bei TIME wird die durschnittliche Zeit für das Verweilen in einem Status berechnet. Beispiel ist die durchschnittliche Leuchtzeit einer Lampe.

Bei CALC wird der vom definierten Datenpunkt gelieferte Wert als Durchnittswert berechnet. Beispiel ist die Messung der durchschnittlichen Aussentemperatur.

Die Berechnung des Durschnittes ist nach der Berechnungsmethode "gleitender Durchschnitt" realisiert. Dabei werden die gemessenen Werte addiert und durch die Anzahl der gemessenen Werte dividiert.

Eine Ausnahme bildet der Schwellwert, der in Tabelle "special" Position 10 definiert werden kann. Definition Schwellwert: ist der Wert (ohne Umrechnung) der für die Berechnung herangezogen wird, wenn der gemessene Wert nicht "kleiner gleich" ist. Beispiel für die Nutzung des Schwellwertes. Für die Einschaltzeit einer Lampe soll die Durschnittsberechnung verwendet werden. Wenn aber die Lampe weniger als 1 Minute eingeschaltet werden, dann soll der Wert nicht verwendet werden, um große Ausreißer zu vermeiden. In diesem Fall würde man den Wert "60000" (ohne Umrechnung heißt hier Millisekunden) in das Feld "Schwellwert" eintragen

2.4.11 Planungsfunktion

Die Planungsfunktion ist eine Alternative zum Trigger. Das heißt, wenn es nicht gewünscht ist auf Änderungen eines Datenpunktes zu reagieren, sondern aufgrund eines geplanten Zeitpunktes, dann muss in der Planungsfunktion die Planung per CRON Syntax eingestellt werden. Die Planungsfunktion ist ausgeschaltet für die Method TIME, da dies nicht sinnvoll erscheint. Grund ist, dass TIME eine Zeitmessung initiiert, wenn ein Status eines Datenpunktes gemeldet wird (z.B. Lampe ist eingeschaltet). TIME ist also auf den Trigger angewiesen, um die Messung zu starten. Die Methoden Delta, DeltaM und Calc sind hingegen nicht auf Trigger (Änderung von Datenpunkten) angewiesen.

Die Planungsfunktion wird aktiviert, wenn ein CRON Expression eingegeben wird. Gleichzeitig wird der Trigger ausgeschaltet.

Es findet keine Syntax-Prüfung statt. Daher empfiehlt es sich einen Schedule Simulator zu nutzen, um evtl. Syntax Fehler zu vermeiden. Beispiele sind:

- Generierung von CRON Expressions aufgrund von Vorgaben: http://crontab-generator.org/

- Ueberpruefung von Cron Expressions: http://crontab.guru/

Beispiele für gültige CRON Expressions

Alle 10 Minuten:	"*/10 * * * *"
Alle 2 Stunden:	"* */2 * * *"
Einmal am Tag um 12:05 Uhr mittags:	"5 12 * * *"
Am 31.12. um 23:59 eines jeden Jahres:	"59 23 31 12 *"

2.4.12 Mehrfachstatus Meldungen zulassen

In der Regel werden Statusänderungen bei Homematic Geräten nur gemeldet, wenn sich der Status tatsächlich geändert hat, also z.B. von true auf false gemeldet wird.

Es gibt nun einige Datenpunkte, die aber regelmässig meldungen senden auch, wenn der Status dabei nicht geändert wird. Das hat z.B. bei einer Zeitmessung theoretisch die Konsequenz, dass die Zeitmessung beendet wird und gleichzeitig die nächste Messung gestartet wird.

Programmtechnisch ist dies abgefangen und es beginnt erst eine Messung, wenn es tatsächlich einen Statuswechsel gegeben hat.

Falls es aber gewünscht ist jede Meldung, und zwar unabhängig von einer Statusänderungen zu loggen, dann kann in Tabelle "special" Spalte 12 "Mehrfachstatus" ein "true" eingetragen. Default ist fault.

2.4.13 Selektives Logging

Es gibt die Möglichkeit alle Änderungen in eine CSV Datei zu loggen. Dies ist aber auf die Dauer nicht zu empfehlen, da erstens die Datei groß wird aber, wichtiger, auch mögliche Delays und unerwünschte Nebenwirkungen auftreten könnten. (es wird synchron geloggt)

Falls es gewünscht ist bestimmte Datenpunkte zu loggen, dann kann dies in Tabelle "special" Spalte 14 "selektives Logging" eingestellt werden. Damit werden lediglich die Daten des ausgewählen Datenpunktes in die CSV Datei geschrieben

2.4.14 Pausieren von Updates

Wenn Änderungen vorgenommen werden, dann kann es sinnvoll sein sämtliche Updates zu stoppen. Ansonsten kann es zu Fehlermeldungen im Log kommen. Das kann erzwungen werden in dem das stopp Kennzeichen auf true gesetzt werden. Es werden keinerlei Updates mehr durchgeführt.

Einstellungen:
Gruppen-Tabelle: Position 19. - true = keine Updates, false = Updates werden durchgeführt

2.4.15 Löschen von Datenpunkten

Datenpunkte werden automatisch angelegt - je nach den Einstellungen der Tabellen. Bei der Definition der Tabellen kann es vorkommen, dass z.B. Namen der Strukturen anders vergeben werden sollen. Dies erfordert eine Löschung der vorhandenen Datenpunkte und die Neuanlage der neuen Datenpunkte. Die Vorgehensweise ist dabei zunächst mal die nicht benötigten Datenpunkte zu löschen und dann erst die Definitionen der neuen Datenpunkte bzw. die neue Konfiguration.

Einstellungen:
Gruppen-Tabelle: Position 19. - true = keine Updates, false = Updates werden durchgeführt. Dies ist die Voraussetzung, dass Datenpunkte löschbar sind

Gruppen-Tabelle: Position 20. - true = die zugehörigen Datenpunkte werden nach Speicherung gelöscht, wenn auch Position 19 auf true gesetzt wurde

2.4.16 Logging

Um nachzuvollziehen welche Updates vorgenommen wurden gibt es zwei Logging-Dateien. Eine Datei zeigt alle Änderungen, die durch Updates der zu loggenden Datenpunkte erzeugt wurden, z.B. Lampe aus - Lampe an etc. Das Logging erfolgt synchron und sollte daher genutzt werden, wenn debugged werden soll. Andererseits wurden bisher noch keine Einschränkungen festgestellt, auch bei längerfristig eingeschalteten Logging.

Ein zweites Log wird immer dann fortgeschrieben, wenn die Perioden geschlossen werden. Dies geschieht i.d.R. kurz nach Mitternacht (einstellbar)

Bei den Einstellungen muss der Pfad eingestellt werden. Ansonsten wird es Fehler im ioBroker LOG nach sich ziehen. Der Pfad ist für eine Standard Installation eines Raspi voreingestellt. Für Windows oder IOS müssen die Pfade angepasst werden

Einstellungen:
var LogPath = "/opt/iobroker/iobroker-data/BSZExtLog.csv";
var TimeLogPath = "/opt/iobroker/iobroker-data/BSZExtTimeLog.csv";

wobei LogPath den Namen der Datei enthält um die regelmäßigen Updates zu loggen und TimeLogPath enthält den Periodenabschluss

3 Beispiele zur Einstellung

Im folgenden werden Beispiele gezeigt wie die Tabellen für bestimmte Aufgaben eingestellt werden können. Die Kombinatorik der Tabellen mit deren Einstellungen ist allerdings sehr groß, so dass nur einige typische Kombinationen gezeigt werden, womit aber ein großer Teil der Möglichkeiten angerissen wird.

Die Vorgehensweise ist die folgende

- Darstellung der Aufgabe

- Umreißen der Lösung der Aufgabe

- Beschreibung der Einstellungen für die drei Tabellen in beschreibender Form und anhand der Tabellen

3.1 Einfache Betriebszeit

Die einfachste Form der Betriebstundenmessung misst einen Verbraucher sobald dieser auf true geschaltet wird bis zum Moment an dem der Verbraucher auf false gesetzt wird. (Beispiel Lampe oder TV). Die Darstellung erfolgt dann durch ddd:hh:mm:ss: Beispiel: der Verbraucher war eingeschaltet; 002:12:05:10, übersetzt 2 Tage 12 Stunden 5 Minuten und 10 Sekunden.

Anmerkung: Die einfachste Einstellung im Programm ist das Hinterlegen eines Datenpunktes für den die Zeit für den Zustand AN (true) gemessen wird. Um dieses zu tun muss lediglich in der Tabelle "Gruppen" an Pos1 der Datenpunkt hinterlegt werden. Weitere Einstellungen werden durch Default-Werte abgeleitet.

Einstellungen aus dem Beispiel:

-Gruppe0 - Messung Betriebsstunden Lampe

Hinzugefügt werden Periodenwerte aber keine Historie. Switches sind eingeschaltet. Somit kann die Anzahl der Ein/Ausschaltungen verfolgt werden. Die Methode ist TIME, da es sich hier um eine Zeitmessung handelt. In Tabelle "Gruppen" Position 9 ist nichts eingetragen. Per default wird dann von true ausgegangen. Das heißt die Messung beginnt, wenn der Status des Gerätes auf true gesetzt wurde.

Das Thema in der Struktur ist "LICHT " (Tabelle Gruppen - Position 2) . Der Name auf Level 5 in der Struktur ist "WZ-Stehlampe " (Tabelle "Logname" Status 1)

In Tabelle Special sind keine Werte eingetragen. Damit ist die Anzeige per default ddd.hh:mm:ss.

Tabelle Gruppen

```
                    1.Homematic ID,                    2.Feldname(no spaces)  3.History    4.DAY  5.Week  6.Month 7.Year  8.Switch   9 - 18 Status to log                                19.stop 20.Loesch

Gruppen[0] = ['hm-rpc.0.JEQ0036841.1.STATE' , 'LICHT'          ,false      ,true   ,true   ,true   ,true   ,true      ,"    ,"    ,"    ,"    ,"    ,"    ,"    ,"    ,"    ,"    ,false ,false];
```

Tabelle Logname

```
                        Stat1      Stat2  Stat3  Stat4        Stat5     Stat6     Stat7    Stat8 Stat9  Stat10

logname[0] =   ['WZ-Stehlampe ,"      ,"      ,"          ,"      ,"      ,"      ,"    ,"    ,"];
```

Tabelle Special

```
//                1.Round          2.add1   3.Faktor          4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze   8.Warten auf Bestaetigung                  9.Durchschnitt -Schwellwert 11 Schedule
special[0] =  ["   ,"              ,"          ,"                    ,"                ,"                ,"                  ,"                                  ,"               ,","];
```

3.2 Betriebszeit mit multiplen Status - Darstellung in Stunden

Nach dem gleichen Prinzip wie die einfache Betriebszeitenmessung erfolgt die Messung, wenn ein Gerät nicht nur ein Status sendet sondern mehrere. Beispiel hier ist eine Heizung, die über Modbus angeschlossen ist und von Start bis Idle durch verschiedene Status läuft. Die Aufgabe ist für jeden Status die Zeiten zu messen und in Stunden darzustellen.

Einstellungen aus dem Beispiel:

-Gruppe0 - Messung Betriebsstunden je Status der Heizung

Hinzugefügt werden Periodenwerte aber keine Historie. Switches sind ausgeschaltet. Die Methode ist TIME, da es sich hier um eine Zeitmessung handelt. In Tabelle "Gruppen" Position 9 - 15 sind die jeweiligen Status eingetragen, die aus dem Modbus Adapter gemeldet werden. Das heißt die Messung beginnt, wenn der jeweilige Status des Gerätes auf gesetzt wurde.

Das Thema in der Struktur ist "Heizung" (Tabelle Gruppen - Position 2) . Alle Status die in der Gruppentabelle angegeben wurden werden separat geloggt.

Allerdings sind die Namen der Status nicht sprechend. Daher können die technischen Namen übersteuert werden mit sprechenden Namen. In diesem Fall: Start, Zuendung , Softstart , Leistungsbrand, Nachlauf, Saugen und Idle. Diese Namen erscheinen nun in Level5 der Datenstruktur (Tabelle "Logname" Status 1 bis Status 7)

In Tabelle Special ist ein Divisor eingetragen von 3.600.000. Dieses stellt die Umrechnung von Millisekunden in Stunden dar. (Tabelle "special" Position 4). Der errechnete Wert wird dann auf zwei Stellen gerundet. . (Tabelle "special" Position 1).

Tabelle Gruppen

```
                    1.Homematic ID,                                    2.Feldname(no spaces)        3.History 4.DAY 5.Week 6.Month 7.Year 8.Switch   9 - 18 Status to log                                  19.stop 20.Loesch
Gruppen[6] = ['modbus.0.holdingRegisters.26_FA1_STATE',  'Heizung'                                     ,false ,true  ,true  ,true   ,true  ,false   ,'1'  ,'2'  ,'3'  ,'4'  ,'5'  ,'7'  ,'99' ,''   ,''   ,''   ,false ,false];
```

Tabelle Logname

```
             Stat1          Stat2          Stat3          Stat4                Stat5          Stat6          Stat7   Stat8 Stat9  Stat10
logname[6] = ['Start'      ,'Zuendung' ,'Softstart','Leistungsbrand'   ,'Nachlauf ,'Saugen'   ,'Idle'   ,''   ,''      ,''];
```

Tabelle Special

```
//                1.Round          2.add1  3.Faktor          4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze   8.Warten auf Bestaetigung                  9.Durchschnitt -Schwellwert 11 Schedule
special[0] =  ['2'   ,''               ,''          ,'3600000'          ,''                ,''                 ,''                 ,''                                        ,''                ,'','"];
```

3.3 Durschnittliche Temperaturen loggen (24-Stunden)

-Gruppe0 - Messung Aussentemperatur mit Durschnittsbildung

Hinzugefügt werden Periodenwerte und Historie. Switches sind nicht notwendig. Die Methode ist CALC, da es sich hier um eine Wertemessung eines Sensors handelt. In Tabelle "Gruppen" Position 9 ist daher CALC eingetragen. Das Thema in der Struktur ist "Temperaturen " (Tabelle Gruppen - Position 2) . Der Name auf Level 5 in der Struktur ist "Aussen24h " (Tabelle "Logname" Status 1)

In Tabelle Special inPosition 1 wird 2 eingetragen, um die Temperaturwerte auf 2 Nachkommastellen zu begrenzen. Weiterhin ist in special 9 ein "true" eingetrage. Damit werden die notwendigen Datenpunkte für die Durschnittsbildung angelegt.

Tabelle Gruppen

```
               1.Homematic ID,            2.Feldname(no spaces)      3.History     4.DAY  5.Week  6.Month 7.Year  8.Switch   9 - 18 Status to log                 19.stop 20.Loesch

Gruppen[0] = [hm-rpc.0.JEQ0014148.1.TEMPERATURE', 'Temperaturen      ,true         ,true  ,true  ,true  ,true  ,false    ,'calc'  ,''  ,''  ,''  ,''  ,''  ,''  ,''  ,''  ,''   ,false ,false];
```

Tabelle Logname

```
                    Stat1     Stat2  Stat3  Stat4      Stat5   Stat6   Stat7  Stat8 Stat9 Stat10

logname[0] =  ['Aussen24h'   ,''     ,''     ,''         ,''     ,''     ,''     ,''   ,''   ,''];
```

Tabelle Special

```
//              1.Round       2.add1  3.Faktor       4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze   8.Warten auf Bestaetigung        9.Durchschnitt -Schwellwert 11 Schedule
special[0] =  [  '2'          ,''      ,''                ,''       ,''          ,''            ,''                          ,''                          ,'true'          ,''                ,''];
```

3.4 Durschnittliche Temperaturen loggen (1 x täglich)

-Gruppe0 - Messung Aussentemperatur mit Durschnittsbildung bei einmaliger Messung pro Tag

Hinzugefügt werden Periodenwerte und Historie. Switches sind nicht notwendig. Die Methode ist CALC, da es sich hier um eine Wertemessung eines Sensors handelt. In Tabelle "Gruppen" Position 9 ist daher CALC eingetragen. Das Thema in der Struktur ist "Temperaturen " (Tabelle Gruppen - Position 2) . Der Name auf Level 5 in der Struktur ist "AussenNoon " (Tabelle "Logname" Status 1)

In Tabelle Special inPosition 1 wird 2 eingetragen, um die Temperaturwerte auf 2 Nachkommastellen zu begrenzen. Weiterhin ist in special 9 ein "true" eingetrage. Damit werden die notwendigen Datenpunkte für die Durchnittsbildung angelegt.

Damit nicht bei jeder vom Geraet gemessenen Temperatur eine Durchnittsrechnung erfolgt wird in special Postion 9 der gewünschte schedule eingetragen. In diesem Fall: '5 12 * * *'

Tabelle Gruppen

```
                1.Homematic ID,                   2.Feldname(no spaces)       3.History      4.DAY  5.Week  6.Month 7.Year  8.Switch   9 - 18 Status to log                      19.stop 20.Loesch
Gruppen[0] = [hm-rpc.0.JEQ0014148.1.TEMPERATURE', 'Temperaturen      ,true          ,true   ,true   ,true   ,true   ,false    ,'calc'  ,''  ,''  ,''  ,''  ,''  ,''  ,''  ,''  ,''   ,false ,false];
```

Tabelle Logname

```
                      Stat1    Stat2  Stat3  Stat4     Stat5   Stat6   Stat7  Stat8  Stat9  Stat10
logname[0] =  ['Aussen24h'   ,''     ,''     ,''        ,''     ,''     ,''     ,''   ,''    ,''];
```

Tabelle Special

```
//              1.Round       2.add1  3.Faktor      4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze   8.Warten auf Bestaetigung      9.Durchschnitt -Schwellwert 11 Schedule
special[0] =  [  '2'          ,''      ,''            ,''        ,''      ,''                ,''                    ,''                             ,'true'        ,''          ,'5 12 * * *'];
```

3.5 Strom/Gas/Wasser-Verbrauchsrechnung

Die Verbrauchsmessung basiert auf Wertemessungen eines Sensors. Dabei wird ein Wert zur Startzeit ermittelt und dann nachdem die Ende-Zeit erreicht wurde wird die Differenz gebildet. Zur Verfügung steht dann also ein Differenzwert. Der Differenzwert wird in den Perioden kumuliert. Der Gesamtwert wird auf Level5 angezeigt.

Somit handelt es sich um die Methode DELTA.

Für die Homematic Energiesensoren ist die Differenz = WATT.
Für Wasser- und Gasuhren ist die Differenz = m3.

Einstellungen aus dem Beispiel:

-Gruppe0 - Hier wird der der Stromverbrauch in kWh gezeigt.

Hinzugefügt werden Periodenwerte aber keine Historie. Switches sind eingeschaltet. Somit kann die Anzahl der ermittelten Messungen vom Energiesensor verfolgt werden. Die Methode ist DELTA, da in Tabelle "Gruppen" - Position 9 "delta" eingegeben wurde

Das Thema in der Struktur ist "Strom " (Tabelle Gruppen - Position 2) . Der Name auf Level 5 in der Struktur ist " kWH-Oekofen " (Tabelle "Logname" Status 1)

Die Berechnung ist Watt / 1000 = kWh gerundet auf 3 Stellen.

(Tabelle Special Position 1 für die Rundung, Der gemessene Wert wird durch Position 3 mit 0,001 multipliziert was Wh zu kWh umrechnet)

Tabelle Gruppen

```
                 1.Homematic ID,                  2.Feldname(no spaces)                3.History  4.DAY  5.Week  6.Month 7.Year  8.Switch   9 - 18 Status to log                    19.stop 20.Loesch

Gruppen[0] = ['hm-rpc.0.KEQ0965841.2.ENERGY_COUNTER', 'Strom'                 ,false  ,true   ,true   ,true   ,true   ,true   ,'delta' ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,false ,false]; // kWh Oekofen
```

Tabelle Logname

```
                       Stat1      Stat2  Stat3  Stat4        Stat5    Stat6    Stat7   Stat8  Stat9  Stat10

logname[0] =  ['kWH-Oekofen' ,''      ,''      ,''           ,''      ,''      ,''      ,''   ,''   ,''];
```

Tabelle Special

```
//                1.Round          2.add1   3.Faktor          4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze   8.Warten auf Bestaetigung                9.Durchschnitt -Schwellwert 11 Schedule
special[0] =  ['3'   ,''    ,'0.001'          ,''          ,''                 ,''                 ,''                                   ,''                ,'',''];  // kwH Oekofen
```

-Gruppe1 - Hier wird der der Stromverbrauch in Euro gezeigt.

Hinzugefügt werden Periodenwerte aber und auch Historie. Switches sind eingeschaltet. Somit kann die Anzahl der ermittelten Messungen vom Energiesensor verfolgt werden. Die Methode ist DELTA, da in Tabelle "Gruppen" - Position 9 "delta" eingegeben wurde

Das Thema in der Struktur ist "Strom " (Tabelle Gruppen - Position 2) . Der Name auf Level 5 in der Struktur ist " EURO-Oekofen " (Tabelle "Logname" Status 1)

Die Berechnung ist Watt / 1000 = kWh gerundet auf 3 Stellen.

Tabelle "Special" Position 1 für die Rundung 2. (Hinweis: "0" würde auf volle Euro runden. Der gemessene Wert wird durch Position 4 durch 1000 dividiert und mit dem kWh Preis multipliziert (0,29) den kWh Preis in Euro zum Ergebnis hat)

Hinweis: in Tabelle "Special" Position 8 ist nichts eingetragen. Somit wird ein Wert von 100 gesetzt. Das bedeutet, dass eine Abweichung bis 100 Watt zwischen 2 Messungen (negativer Verbrauch) akzeptiert wird. Ein Wert darüber hinaus wird nicht als Verbrauch gewertet. (Rücksetzen des Energiesensors

Tabelle Gruppen

```
                1.Homematic ID,                    2.Feldname(no spaces)                3.History 4.DAY 5.Week 6.Month 7.Year 8.Switch  9 - 18 Status to log                    19.stop 20.Loesch

Gruppen[1] = ['hm-rpc.0.KEQ0965841.2.ENERGY_COUNTER', 'Strom'                ,true ,true ,true ,true ,true ,true ,'delta' ,''   ,''   ,''   ,''   ,''   ,''   ,''   ,''   ,false ,false]; // Euro Oekofen
```

Tabelle Logname

```
                        Stat1      Stat2 Stat3 Stat4       Stat5    Stat6    Stat7   Stat8 Stat9 Stat10

logname[1] =  ['EURO-Oekofen' ,''     ,''     ,''          ,''     ,''     ,''     ,''   ,''   ,''];
```

Tabelle Special

```
//                1.Round          2.add1  3.Faktor          4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze   8.Warten auf Bestaetigung                9.Durchschnitt -Schwellwert 11 Schedule
special[1] =  ['2'      ,''          ,'0.29'          ,'1000'          ,''                ,''                ,''                                ,''                ,''                ,'','']; // kwH in Euro fuer Oekofen
```

Gesamtübersicht der Einstellungen für die Stromverbrauchsrechnung

Tabelle Gruppen

```
                1.Homematic ID,                    2.Feldname(no spaces)                3.History 4.DAY 5.Week 6.Month 7.Year 8.Switch  9 - 18 Status to log                    19.stop 20.Loesch

Gruppen[0] = ['hm-rpc.0.KEQ0965841.2.ENERGY_COUNTER', 'Strom'                ,false ,true ,true ,true ,true ,true ,'delta' ,''   ,''   ,''   ,''   ,''   ,''   ,''   ,''   ,false ,false]; // kWh Oekofen

Gruppen[1] = ['hm-rpc.0.KEQ0965841.2.ENERGY_COUNTER', 'Strom'                ,true ,true ,true ,true ,true ,true ,'delta' ,''   ,''   ,''   ,''   ,''   ,''   ,''   ,''   ,false ,false]; // Euro Oekofen
```

Tabelle Logname

```
                        Stat1      Stat2 Stat3 Stat4       Stat5    Stat6    Stat7   Stat8 Stat9 Stat10

logname[0] =  ['kWH-Oekofen' ,''     ,''     ,''          ,''     ,''     ,''     ,''   ,''   ,''];

logname[1] =  ['EURO-Oekofen' ,''     ,''     ,''          ,''     ,''     ,''     ,''   ,''   ,''];
```

Tabelle Special

```
//                1.Round          2.add1  3.Faktor          4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze   8.Warten auf Bestaetigung                9.Durchschnitt -Schwellwert 11 Schedule
special[0] =  ['3'      ,''          ,'0.001'         ,''                ,''                ,''                ,''                                ,''                ,''                ,'','']; // kwH Oekofen

special[1] =  ['2'      ,''          ,'0.29'          ,'1000'          ,''                ,''                ,''                                ,''                ,''                ,'','']; // kwH in Euro fuer Oekofen
```

3.6 Tankstandmessungen

Die Tankstandmessung basiert auf Wertemessungen eines Level-Mess-Sensors, der einen Tankstand mittels modbus, S7 etc. an ioBroker übermittelt . Dabei wird ein Wert zur Startzeit ermittelt und dann nachdem die Ende-Zeit erreicht wurde wird die Differenz gebildet. Zur Verfügung steht dann also ein Differenzwert. Der Differenzwert wird in den Perioden kumuliert. Der Gesamtwert wird auf Level5 angezeigt. Der Gesamtwert nimmt stetig ab

Somit handelt es sich um die Methode DELTAM.

Der gemessene Wert wird in Liter übermittelt. Eine Besonderheit hier (ähnlich wie bei Stromverbrauchsmessungen nur regelmäßiger) ist, dass die gemeldete Literzahl nicht stetig abnehmend ist, sondern mal ein paar Liter mehr oder ein paar Liter weniger gemessen werden. (Ungenauigkeiten im Messverfahren)

.

Einstellungen aus dem Beispiel:

-Gruppe0 - hier wird der Tankstand/Oelverbrauch gemessen.

Es soll kontinuierlich der Tankstand gemessen werden. Dabei sollen aber die Differenzen als Verbrauch berechnet werden.

Das Thema in der Struktur ist "OEL " (Tabelle Gruppen - Position 2) . Der Name auf Level 5 in der Struktur ist " OelRest " (Tabelle "Logname" Status 1)

Hinzugefügt werden Periodenwerte (Tabelle "Gruppen" Position 4 - 7) und auch Historie wird aktiviert (Tabelle "Gruppen" Position 7) aber keine Switches. (Tabelle "Gruppen" Position 8)

Die Methode ist DELTAM (Tabelle "Gruppen" Position 9)

Es wird keine besondere Umrechnung definiert, also ist die Darstellung in Liter, wie vom Adapter gesendet. Allerdings wird der Wert auf 2 Nachkommastellen gerundet (Tabelle "special" - Position 1)

Die Delta-Grenze ist auf 500 Liter gesetzt. Das heisst, wenn der Tankstand sich um mehr als 500 Liter erhöht, dass dies ein Tankvorgang ist. Somit werden die hinzugefügten Liter nicht als negativer Verbrauch berechnet. (Tabelle "special" - Position 7).

Gemessen wird ohne, dass eine Bestätigung des gemessenen Wertes erwartet wird. (Tabelle "special" - Position 8)

Tabelle Gruppen

```
                    1.Homematic ID,                    2.Feldname(no spaces)          3.History 4.DAY 5.Week 6.Month 7.Year 8.Switch  9 - 18 Status to log                                    19.stop 20.Loesch
Gruppen[0] = ['javascript.0.BSZ.Oelstand'   , 'OEL'                                    ,true  ,true  ,true  ,true  ,true  ,false ,'deltam','' ,'' ,'' ,'' ,'' ,'' ,'' ,'' ,'' ,'' , false,false];
```

Tabelle Logname

```
                        Stat1      Stat2  Stat3  Stat4       Stat5    Stat6     Stat7   Stat8 Stat9  Stat10
logname[0] =   ['OELRest'   ,''      ,''     ,''          ,''      ,''      ,''      ,''   ,''   ,''];
```

Tabelle Special

```
//                  1.Round          2.add1  3.Faktor          4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze   8.Warten auf Bestaetigung                  9.Durchschnitt -Schwellwert 11 Schedule
special[0] =  ['2'  ,''              ,''             ,''                  ,''              ,''                ,'500'              ,'false'                          ,''              ,'','']; // DeltaM
```

-Gruppe1 - hier wird der Tankstand/Verbrauch bewertet in Euro dargestellt

Es soll kontinuierlich der Tankstand gemessen werden. Dabei sollen aber die Differenzen als Verbrauch berechnet werden. Diese Differenzen sollen in Euro dargestellt werden. Der gesamte Öl-Vorrat wird ebenfalls in Euro bewertet

Das Thema in der Struktur ist "OEL " (Tabelle Gruppen - Position 2) . Der Name auf Level 5 in der Struktur ist " OelEuro " (Tabelle "Logname" Status 1)

Hinzugefügt werden Periodenwerte (Tabelle "Gruppen" Position 4 - 7) aber keine Historie wird aktiviert (Tabelle "Gruppen" Position 7) und auch keine Switches. (Tabelle "Gruppen" Position 8)

Die Methode ist DELTAM (Tabelle "Gruppen" Position 9)

Es wird eine Umrechnung mit 40 Cent je Liter definiert (Tabelle "special" - Position 3), also ist die Darstellung in Euro. Der Wert auf 2 Nachkommastellen gerundet (Tabelle "special" - Position 1)

Die Delta-Grenze ist auf 500 Liter gesetzt. Das heisst, wenn der Tankstand sich um mehr als 500 Liter erhöht, dass dies ein Tankvorgang ist. Somit werden die hinzugefügten Liter nicht als negativer Verbrauch berechnet. (Tabelle "special" - Position 7).

Gemessen wird ohne, dass eine Bestätigung des gemessenen Wertes erwartet wird. (Tabelle "special" - Position 8)

Tabelle Gruppen

```
                    1.Homematic ID,                    2.Feldname(no spaces)          3.History 4.DAY 5.Week 6.Month 7.Year 8.Switch  9 - 18 Status to log                                  19.stop 20.Loesch
Gruppen[1] = ['javascript.0.BSZ.Oelstand'  , 'OEL'                                      ,false ,true  ,true  ,true  ,true  ,false ,'deltam','"   ,"   ,"   ,"   ,"   ,"   ,"   ,"   ,"   , false,false];
```

Tabelle Logname

```
                      Stat1      Stat2 Stat3 Stat4       Stat5     Stat6     Stat7    Stat8 Stat9 Stat10
logname[1] =  ['OELEuro'   ,''      ,''     ,''         ,''       ,''       ,''      ,''   ,''   ,''];
```

Tabelle Special

```
//                1.Round          2.add1  3.Faktor          4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze   8.Warten auf Bestaetigung                  9.Durchschnitt -Schwellwert 11 Schedule
special[1] =  ['2'   ,''               ,'0.4'          ,''                    ,''                 ,''                  ,'500'            ,'false'                                ,''              ,'','']; // DeltaM
```

Gesamtübersicht der Einstellungen für die Tankstandmessung

Tabelle Gruppen

```
                   1.Homematic ID,                       2.Feldname(no spaces)            3.History  4.DAY  5.Week  6.Month 7.Year  8.Switch   9 - 18 Status to log                              19.stop 20.Loesch
Gruppen[0] = ['javascript.0.BSZ.Oelstand'   , 'OEL'                                        ,true   ,true   ,true   ,true   ,true   ,false  ,'deltam',''   ,''   ,''   ,''   ,''   ,''   ,''   ,''   ,''    , false,false];
Gruppen[1] = ['javascript.0.BSZ.Oelstand'   , 'OEL'                                        ,false  ,true   ,true   ,true   ,true   ,false  ,'deltam',''   ,''   ,''   ,''   ,''   ,''   ,''   ,''   ,''    , false,false];
```

Tabelle Logname

```
                    Stat1      Stat2  Stat3  Stat4        Stat5    Stat6    Stat7   Stat8  Stat9  Stat10
logname[0] =   ['OELRest'    ,''      ,''      ,''           ,''      ,''      ,''      ,''    ,''    ,''];

logname[1] =   ['OELEuro'    ,''      ,''      ,''           ,''      ,''      ,''      ,''    ,''    ,''];
```

Tabelle Special

```
//                1.Round         2.add1   3.Faktor          4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze    8.Warten auf Bestaetigung                  9.Durchschnitt -Schwellwert 11 Schedule
special[0] =   ['2'    ,''              ,''              ,''                  ,''                 ,'500'            ,'false'                              ,''                ,'',''];  // DeltaM
special[1] =   ['2'    ,''              ,'0.1'           ,''                  ,''                 ,'500'            ,'false'                              ,''                ,'',''];  // DeltaM
```

3.7 Pelletverbrauchsmessung

Die Pelletverbrauchsmessung basiert auf Zeitmessungen. Somit ist die Methode TIME. Dabei wird eine Startzeit in Millisekunden ermittelt und dann nachdem die Ende-Zeit festgestellt wurde die Differenz gebildet. Zur Verfügung steht dann also die Laufzeit in Millisekunden.

Einstellungen aus dem Beispiel:

-Gruppe0 . hier wird die Pelletschneckenlaufzeit erfasst.

Hinzugefügt werden Periodenwerte aber keine Historie und keine Switches. Gemessen wird, wenn das Gerät auf Status "false" wechselt. Die Methode ist TIME, da keine andere Methode in Position 9 angegeben wurde

Das Thema in der Struktur ist "Pellet" (Position 2) . Der Name auf Level 5 in der Struktur ist "Schneckenzeit" (Tabelle "Logname" Status 1)

Es werden keine Umrechnungen vorgenommen. Daher ist die Darstellung ddd:hh:mm:ss (siehe Tabelle "Special")

Tabelle Gruppen

```
                    1.Homematic ID,                    2.Feldname(no spaces)        3.History  4.DAY  5.Week  6.Month 7.Year  8.Switch   9 - 18 Status to log                    19.stop 20.Loesch

Gruppen[0]  = ['hm-rpc.0.KEQ0178063.1.STATE'      ,'Pellet'              ,false  ,true   ,true   ,true   ,true   ,true   ,'false' ,"    ,"    ,"    ,"    ,"    ,"    ,"    ,"    ,"    ,false ,false]; // Schneckenzeit
```

Tabelle Logname

```
                        Stat1      Stat2  Stat3  Stat4       Stat5    Stat6    Stat7   Stat8 Stat9  Stat10

logame[0]  =  ['Schneckenzeit',"      ,"      ,"          ,"      ,"      ,"      ,"    ,"    ,"    ,"];
```

Tabelle Special

```
//                1.Round        2.add1  3.Faktor        4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze   8.Warten auf Bestaetigung                9.Durchschnitt -Schwellwert 11 Schedule
special[0]  =  ["                ,"              ,"          ,"          ,"          ,"              ,"                  ,"                                  ,"                ,","]; // Schneckenzeit
```

- Gruppe1 - Die Zeit, die die Schnecke läuft ist Ausgang für die Berechnung des Pelletverbrauches.

Es wird eine Historie angelegt (Tabelle Gruppen Position 3). Die Perioden DAY, WEEK,MONTH und YEAR sollen geführt werden (Tabelle Gruppen Position 4 - 7). Die Anzahl der Läufe wird nicht dokumentiert (Tabelle Gruppen Position 8). Die Zeit wird gemessen, wenn der Status des Gerätes false ist (Tabelle Gruppen Positon 9).

Der sichtbare Name in der Struktur auf Level 5 ist "PelletKG" (Tabelle Logname - Status1)

Die Berechnung ist 7,5 KG Pellets pro Minute Schneckenlaufzeit. Dabei soll das Ergebnis auf 2 Stellen gerundet werden. (Tabelle Special Position 1 für die Rundung, Position 3 für 7,5 KG und Position 4 für die Umrechnung auf Minute)

Tabelle Gruppen

```
                1.Homematic ID,                    2.Feldname(no spaces)        3.History  4.DAY  5.Week  6.Month 7.Year  8.Switch   9 - 18 Status to log                                19.stop 20.Loesch

Gruppen[1]  = ['hm-rpc.0.KEQ0178063.1.STATE'       ,'Pellet'                    ,true  ,true  ,true  ,true  ,true  ,false  ,'false' ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,false ,false]; // Kumuliert   KG
```

Tabelle Logname

```
                        Stat1      Stat2  Stat3  Stat4       Stat5     Stat6     Stat7    Stat8 Stat9  Stat10

logname[1]  =  ['PelletKG'   ,''      ,''      ,''          ,''      ,''      ,''      ,''    ,''    ,''];
```

Tabelle Special

```
//                1.Round          2.add1   3.Faktor          4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze   8.Warten auf Bestaetigung                  9.Durchschnitt -Schwellwert 11 Schedule
special[1]  =  ['2'               ,''                ,'7.5'         ,'60000'  ,''           ,''                ,''                                ,''                        ,''                ,''];// Pelletverbrauch KG
```

- Gruppe2 - Die Zeit, die die Schnecke läuft ist Ausgang für die Berechnung des Pelletverbrauches bewertet in Euro.

Es wird eine Historie angelegt (Tabelle Gruppen Position 3). Die Perioden DAY, WEEK,MONTH und YEAR sollen geführt werden (Tabelle Gruppen Position 4 - 7). Die Anzahl der Läufe wird nicht dokumentiert (Tabelle Gruppen Position 8). Die Zeit wird gemessen, wenn der Status des Gerätes false ist (Tabelle Gruppen Positon 9).

Der sichtbare Name in der Struktur auf Level 5 ist "PelletEURO" (Tabelle Logname - Status1)

Die Berechnung ist 1,725 Euro pro Minute Schneckenlaufzeit. Das bedeutet, dass der Preis bei 1,725 Euro pro 7,5 KG Pellets liegt. Dabei soll das Ergebnis auf 2 Stellen gerundet werden. (Tabelle Special Position 1 für die Rundung, Position 3 für 1,725 Euro und Position 4 für die Umrechnung auf Minute)

Tabelle Gruppen

```
                1.Homematic ID,                    2.Feldname(no spaces)        3.History  4.DAY  5.Week  6.Month 7.Year  8.Switch   9 - 18 Status to log                                19.stop 20.Loesch

Gruppen[2]  = ['hm-rpc.0.KEQ0178063.1.STATE'      ,'Pellet'                    ,true  ,true  ,true  ,true  ,true  ,false  ,'false' ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,false ,false]; // Kumuliert   Euro
```

Tabelle Logname

```
                        Stat1       Stat2  Stat3  Stat4        Stat5     Stat6     Stat7    Stat8 Stat9  Stat10

logname[2]  =  ['PelletEURO'  ,''      ,''      ,''           ,''      ,''      ,''      ,''    ,''    ,''];
```

Tabelle Special

```
//                1.Round          2.add1  3.Faktor          4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze   8.Warten auf Bestaetigung                  9.Durchschnitt -Schwellwert 11 Schedule
special[2]  =  ['2'                ,''                ,'1.725'      ,'60000'  ,''           ,''                ,''                                  ,''                                ,''                ,'','']; // Pelletverbrauch  €
```

- Gruppe3 berechnet den verbleibenden Silobestand.

Es wird keine Historie angelegt (Tabelle Gruppen Position 3). Die Perioden DAY, WEEK,MONTH und YEAR sollen nicht geführt werden (Tabelle Gruppen Position 4 - 7). Die Anzahl der Läufe wird nicht dokumentiert (Tabelle Gruppen Position 8). Die Zeit wird gemessen, wenn der Status des Gerätes false ist (Tabelle Gruppen Positon 9).

Der sichtbare Name in der Struktur auf Level 5 ist "PelletRest" (Tabelle Logname - Status1)

Die Berechnung ist 7,5 KG Pellets pro Minute Schneckenlaufzeit. Dabei soll das Ergebnis auf 2 Stellen gerundet werden. (Tabelle Special Position 1 für die Rundung, Position 3 für 7,5 KG und Position 4 für die Umrechnung auf Minute). Die Besonderheit hier ist, dass der Wert 7,5 KG als -7,5 angegeben wird. Damit wird fortlaufend subtrahiert.

Damit der Pelletbestand angezeigt wird muss auf Level5 ein Anfangsbestand eingegeben werden, bzw. bei Tankfüllungen die jeweilige Füllung addiert werden. Dies kann in VIS über ein Widget realisiert werden. Alternativ kann natürlich auch in ioBroker unter Zustände manuell ein korrigierter Bestand eingegeben werden

Tabelle Gruppen

```
                    1.Homematic ID,                         2.Feldname(no spaces)          3.History  4.DAY  5.Week  6.Month 7.Year  8.Switch   9 - 18 Status to log                                  19.stop 20.Loesch

Gruppen[3]  = ['hm-rpc.0.KEQ0178063.1.STATE'        ,'Pellet'                  ,false  ,false  ,false  ,false  ,false  ,false  ,'false' ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,false ,false]; // Restbestand KG
```

Tabelle Logname

```
                        Stat1       Stat2  Stat3  Stat4          Stat5     Stat6     Stat7    Stat8 Stat9  Stat10

logname[3]  =  ['PelletRest'  ,''      ,''      ,''            ,''      ,''      ,''      ,''    ,''    ,''];
```

Tabelle Special

```
//                  1.Round          2.add1   3.Faktor          4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze    8.Warten auf Bestaetigung                  9.Durchschnitt -Schwellwert 11 Schedule
special[3]  =  ['2'                 ,''                ,'-7.5'         ,'60000'  ,''           ,''                ,''                                  ,''                        ,''                ,''];  // Pellet Restbestand
```

- Gruppe4 - hält den aktuellen Zustand der Schnecke als Zahl fest

Es wird keine Historie angelegt (Tabelle Gruppen Position 3). Die Perioden DAY, WEEK,MONTH und YEAR sollen nicht geführt werden (Tabelle Gruppen Position 4 - 7). Die Anzahl der Läufe wird nicht dokumentiert (Tabelle Gruppen Position 8). Die Methode ist nun nicht TIME sondern CALC, da ja auch keine Zeitwerte gemessen werden sollen ist (Tabelle Gruppen Positon 9).

Der sichtbare Name in der Struktur auf Level 5 ist "SchneckeSchalt" (Tabelle Logname - Status1)

Die Berechnung ist zum jeweiligen aktuellen booleschen Wert die Zahl 1 zu addieren. Damit wird der Wert 1, wenn die Schnecke aktiv ist und 2, wenn die Schnecke ausgeschaltet ist. (Tabelle Special Position 2)

Tabelle Gruppen

```
                 1.Homematic ID,                    2.Feldname(no spaces)          3.History  4.DAY  5.Week  6.Month 7.Year  8.Switch   9 - 18 Status to log                              19.stop 20.Loesch
Gruppen[4]  = ['hm-rpc.0.KEQ0178063.1.STATE'         ,'Pellet'                    ,false  ,false  ,false  ,false  ,false  ,false  ,'calc'  ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,false ,false]; // Laufzeiten f.Graphik
```

Tabelle Logname

```
                      Stat1      Stat2  Stat3  Stat4      Stat5    Stat6    Stat7   Stat8  Stat9  Stat10
logname[4]  =  ['SchneckeSchalt',''      ,''      ,''          ,''      ,''      ,''      ,''   ,''   ,''];
```

Tabelle Special

```
//               1.Round          2.add1   3.Faktor          4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze    8.Warten auf Bestaetigung              9.Durchschnitt -Schwellwert 11 Schedule
special[4]  =  [''                ,'1'               ,''          ,''       ,''         ,''                ,''                                ,''                    ,'',''];  // für Graphik
```

Gesamtübersicht der Einstellungen für die Pelletverbrauchsrechnung

Tabelle Gruppen

```
                  1.Homematic ID,                   2.Feldname(no spaces)       3.History 4.DAY 5.Week 6.Month 7.Year 8.Switch  9 - 18 Status to log                       19.stop 20.Loesch

Gruppen[0]  = ['hm-rpc.0.KEQ0178063.1.STATE'     ,'Pellet'        ,false ,true  ,true  ,true  ,true  ,true  ,'false' ,''  ,''  ,''  ,''  ,''  ,''  ,''  ,''   ,false ,false]; // Schneckenzeit

Gruppen[1]  = ['hm-rpc.0.KEQ0178063.1.STATE'     ,'Pellet'        ,true  ,true  ,true  ,true  ,true  ,false ,'false' ,''  ,''  ,''  ,''  ,''  ,''  ,''  ,''   ,false ,false]; // Kumuliert   KG

Gruppen[2]  = ['hm-rpc.0.KEQ0178063.1.STATE'     ,'Pellet'        ,true  ,true  ,true  ,true  ,true  ,false ,'false' ,''  ,''  ,''  ,''  ,''  ,''  ,''  ,''   ,false ,false]; // Kumuliert   Euro

Gruppen[3]  = ['hm-rpc.0.KEQ0178063.1.STATE'     ,'Pellet'        ,false ,false ,false ,false ,false ,false ,'false' ,''  ,''  ,''  ,''  ,''  ,''  ,''  ,''   ,false ,false]; // Restbestand KG

Gruppen[4]  = ['hm-rpc.0.KEQ0178063.1.STATE'     ,'Pellet'        ,false ,false ,false ,false ,false ,false ,'calc'  ,''  ,''  ,''  ,''  ,''  ,''  ,''  ,''   ,false ,false]; // Laufzeiten f.Graphik
```

Tabelle Logname

```
                        Stat1        Stat2  Stat3  Stat4      Stat5   Stat6   Stat7  Stat8 Stat9  Stat10

logame[0]  =  ['Schneckenzeit',''     ,''     ,''         ,''     ,''     ,''     ,''   ,''   ,''];

logname[1]  =  ['PelletKG'    ,''     ,''     ,''         ,''     ,''     ,''     ,''   ,''   ,''];

logname[2]  =  ['PelletEURO'  ,''     ,''     ,''         ,''     ,''     ,''     ,''   ,''   ,''];

logname[3]  =  ['PelletRest'  ,''     ,''     ,''         ,''     ,''     ,''     ,''   ,''   ,''];

logname[4]  =  ['SchneckeSchalt',''   ,''     ,''         ,''     ,''     ,''     ,''   ,''   ,''];
```

Tabelle Special

```
//             1.Round        2.add1  3.Faktor       4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze   8.Warten auf Bestaetigung        9.Durchschnitt -Schwellwert 11 Schedule
special[0]  =  [''            ,''             ,''            ,''         ,''          ,''                 ,''                              ,''                ,'',''];  // Schneckenzeit
special[1]  =  ['2'           ,''             ,'7.5'         ,'60000'    ,''          ,''                 ,''                              ,''                ,'',''];  // Pelletverbrauch KG
special[2]  =  ['2'           ,''             ,'1.725'       ,'60000'    ,''          ,''                 ,''                              ,''                ,'',''];  // Pelletverbrauch  €
special[3]  =  ['2'           ,''             ,'-7.5'        ,'60000'    ,''          ,''                 ,''                              ,''                ,'',''];  // Pellet Restbestand
special[4]  =  [''            ,'1'            ,''            ,''         ,''          ,''                 ,''                              ,''                ,'',''];  // für Graphik
```

3.8 Logikerweiterungen

Die Logikerweiterung bietet die Möglichkeit eine eigene Logik zur Berechnung von Werten hinzuzufügen, ohne tief in das Script eingreifen zu müssen. Das hier gezeigte Beispiel nimmt einen Wert aus dem modbus Adapter und berechnet damit einen theoretischen Pelletverbrauch. Hintergrund ist, dass der modbus nur die Saugzeit nicht aber die Pelletlaufzeit sendet. Die Saugzeit hat aber Zeitabschnitte in denen zwar gesaugt, nicht aber gefördert wird. Somit kann die Saugzeit nicht mit einer einfachen Berechnung in KG pro Zeiteinheit umgerechnet werden.

Dennoch basiert der Ansatz hier auf eine Zeitberechnung, d.h., dass die Saugzeit in Millisekunden an die eigene Logik gesendet wird. zurück kommt dann eine Gesamtwert in KG.

Einstellungen aus dem Beispiel:

-Gruppe0 . hier wird die Saugzeit gemessen.

Hinzugefügt werden Periodenwerte aber keine Historie und keine Switches. Gemessen wird, wenn das Gerät auf Status "7" wechselt. Die Methode ist TIME, da keine andere Methode in Tabelle Gruppen - Position 9 angegeben wurde

Das Thema in der Struktur ist "Pellet" (Tabelle Gruppen - Position 2) . Der Name auf Level 5 in der Struktur ist "PelletTheor" (Tabelle "Logname" Status 1)

Die Individual-Funktion heißt "oekofen" (Tabelle Special - Position 6). Das heisst, dass die Funktion vor allen weiteren Berechnungen aufgerufen wird. Zurück kommt in diesem Fall ein KG-Wert

Da der KG Wert (der jetzt von der Individual-Funktion kommt) ein Verbrauch darstellt und der Silobestand angezeigt werden soll, wird der Verbrauch mit -1 multipliziert (Tabelle Special - Position 3). Anschließend wird auf 2 Stellen gerundet (Tabelle Special - Position 1)

Damit der Pelletbestand angezeigt wird muss auf Level5 ein Anfangsbestand eingegeben werden, bzw. bei Tankfüllungen die jeweilige Füllung addiert werden. Dies kann in VIS über ein Widget realisiert werden. Alternativ kann natürlich auch in ioBroker unter Zustände manuell ein korrigierter Bestand eingegeben werden

Tabelle Gruppen

```
                    1.Homematic ID,                    2.Feldname(no spaces)   3.History 4.DAY 5.Week 6.Month 7.Year 8.Switch  9 - 18 Status to log                                  19.stop 20.LoeschGruppen[0] =
['modbus.0.holdingRegisters.26_FA1_STATE' ,'Oekofen'            ,false    ,true  ,true  ,true  ,true  ,false ,'7'   ,''   ,''   ,''   ,''   ,''   ,''   ,''   ,''   ,''   ,''   ,false ,false]; // Oekofen
```

Tabelle Logname

```
                    Stat1      Stat2  Stat3  Stat4     Stat5     Stat6     Stat7  Stat8  Stat9  Stat10
logname[0] =  ['PelletTheor'   ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''    ,''];
```

Tabelle Special

```
//                1.Round          2.add1  3.Faktor          4. Divisor  5.add2  6.Individuallogik  7: DELTA(M)Grenze   8.Warten auf Bestaetigung                  9.Durchschnitt -Schwellwert 11 Schedule
special[0] =  ['2'     ,''        ,'-1'          ,''            ,''      ,'oekofen'          ,''                                ,''                ,''                ,''];// Sonderlogik fuer Saugzeiten Oekofen - theoretischer
Pelletverbrauch
```

4 Programmablauflogik

Für Interessierte ist im folgenden der Ablauf des Programmes dokumentiert. Dabei wurde auch die Dokumentation des Programmes und auch die Reihenfolge der Funktionen entsprechend dieser Beschreibung angepasst.

4.1 Programminitiierung

Die Programminitiierung umfasst die Definition von Variablen (dazu gehören auch die Nutzereinstellungen), die Vorbereitung der Variablen für den Programmablauf wie Plausibilitäts-Check, Einstellen von Default-Werten, Umwandeln von String in Zahlen, etc.

Der nächste Schritt ist das Erstellen der notwendigen Datenpunkte falls noch nicht vorhanden und dem -falls so eingestellt- Löschen von Datenpunkten

Danach werden die Trigger und Schedule eingestellt um zu den geplanten Events die Programmablauflogik zu starten

Die Einbindung von Triggern und Schedule ist eine häufige Form zur Nutzung in JS-Programmen für ioBroker. Dabei bedeutet Trigger, dass auf die Änderung von den definierten Datenpunkten gewartet wird, um dann die Programmablauflogik zu starten. Dieses ist hier durch die Funktion "on({id:" realisiert. Ein Schedule wird im Gegensatz dazu zu bestimmten -ggf wiederkehrenden- Zeiten aufgerufen. Beispiel für die Nutzung im BSZ ist schedule(special[0][10]....

Im Programm ist ein Block "onid", ein Block "schedule" und eine weitere Schedule-Funktion zu finden.

Der "onid"-Block, reagiert auf Datenänderungen.

In manchen Fällen ist es aber gewünscht nicht auf Datenänderungen zu reagieren, sondern nach einem Zeitplan die Ablauflogik zu starten (siehe dazu nähere Erklärungen im Funktionsteil "Planungen". Hierzu dient der "Schedule"-Block.

Ein weiterer Schedule ist eingerichtet, um bei Periodenwechsel ggf. definierte Perioden zurückzusetzen und ggf. Historienwerte zu schreiben.

Anmerkung: der hier beschriebene Programmteil wird nur zu bestimmten Events ausgelöst. Diese sind:
- Das Programm wird verändert und gespeichert (z.B. update der Einstellungen)
- Das Programm wird manuell gestartet (im Reiter "Skripte")
- Die Java-Skript-Instanz startet neu. Das kann unterschiedliche Gründe haben:
  + Die Instanz wird manuell neu gestartet (Reiter Instanzen)
  + ioBroker wird neu gestartet
  + Die Java-Skript Instanz ist abgestürzt und startet automatisch neu, was z.B. durch ein anderes fehlerhaftes Programm verursacht sein kann

Im Programm selber sind die Teile der Programminitiierung so dargestellt:
Part1 Initiierung
Part1.1 Einstellungen
Part1.2 Allgemeine Variablen
Part1.3 Aufbereitung der Tabellen
Part1.4 Anlegen oder Löschen von Datenpunkten
Part 1.4.1 Vorbereiten zum Anlegen/Löschen von Datenpunkten
Part 1.4.2 Ausführen Anlegen/Löschen von Datenpunkten
Part1.5 Definition der Trigger und Schedule

4.2 Update Funktionen

Der Update Teil des Programmes berechnet die Daten entsprechend der Einstellungen, die in den Einstellungstabellen definiert wurden. Die Daten werden in der Objektstruktur der Java-Skript Instanz abgelegt. (Reiter Objekte)

Die Update Funktion ist zunächst grundsätzlich nach den Methoden getrennt. Es gibt also einen Programmteil für CALC, Delta (Delta und DeltaM) und TIME. Dabei ist TIME noch in zwei Teile unterteilt LEVEL-Geräte und NICHT-LEVEL-Geräte. (wird ggf später zusammengeführt)

Die jeweiligen Teile rufen dann dieselben Routinen für das Update der Perioden und auch der Durchschnittsberechnung auf.

Im Programmteil selber sind die Teile der Update-Funktionen so dargestellt:
Part2 Updatefunktionen
Part2.1 Core-Update Funktionen (Methoden CALC,DELTA,TIME)
Part2.2 Kumulation in Perioden schreiben
Part2.3 Durchschnittswerte ermitteln und schreiben
Part2.4 Individualfunktion

4.3 Periodenwechsel

Der Periodenwechsel besteht aus zwei Programmteilen, nämlich der Vorbereitung und der Ausführung des Wechsels. Der Periodenwechsel ist nur dann relevant, wenn überhaupt Perioden oder Historie verwendet werden.

Im Wesentlichen werden folgende Funktionen durchlaufen

- Tageswechsel wird durchgeführt immer, wenn das Programm ausgeführt wird. Es wird also nicht überprüft, ob ein neuer Tag angebrochen ist, sondern entsprechend des Schedules aus der Programminitiierung ausgeführt. Aktion: Nullen der Periode Tag. Schreiben des Wertes aus dem Tag in den Datenpunkte "DAY.BEFORE".

- Wochenwechsel. Das Programm ermittelt ob bei Ausführung der Tag ein Montag ist. In diesem Fall wird die Woche genullt und der vorherige Wert in "WEEK.BEFORE" geschrieben.

-Monatswechsel. Das Programm ermittelt ob das Datum mit "01" beginnt - also dem ersten Tag des Monats. In diesem Fall wird der Monat genullt und der vorherige Wert in "MONTH.BEFORE geschrieben. Wenn Historie aktiv ist wird der Monatswert noch in den abgeschlossenen Monat der Historie geschrieben. z.B. "HISTORY.MONTH.201603"

- Jahreswechsel. das Programm ermittelt ob das Datum mit 01.01. beginnt, also dem ersten Tag des Jahres. In diesem Fall wird das Jahr genullt und der vorherige Wert in "YEAR.BEFORE" geschrieben. Wenn Historie aktiv ist wird der Monatswert noch in das abgeschlossene Jahr der Historie geschrieben. z.B. "HISTORY.YEAR.2016"

Im Programmteil selber sind die Teile der Update-Funktionen so dargestellt:
Part3 Periodenwechsel
Part 3.1 Vorbereitung Periodenwechsel
Part 3.2 Ausführen Periodenwechsel

4.4 Allgemeine Funktionen

In Part4 des Programmes befinden sich allgemeine Funkionen, die mehrfach aus den unterschiedlichen Programmen aufgerufen werden.