

Name	Beschreibung	Bedingung (Wenn...)	Aktivität (Dann..., Sonst...)	Aktion
STROM.Update		Zeit: Periodisch Ganztägig beginnend am 02.09.2019 zu Zeitpunkten auslösen	Skript: ... sofort ausführen	<input type="checkbox"/> systemintern
Bedingung: Wenn...				
Zeitsteuerung				
Zeit: Zeitbereich ganztägig Serienmuster: Zeitintervall, alle 3 Minuten Gültigkeitsdauer: Beginn: 02.09.2019, kein Enddatum				
zu Zeitpunkten auslösen				
ODER				
Aktivität: Dann... <input checked="" type="checkbox"/> Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).				
Skript				
<pre> ! ----- ! STROM.Update1.hms ! ! HomeMatic Skript für HB-ES-S0-CNT6 ! Version 0.22 ! 2019-09-03 Tom Major (Creative Commons) ! https://creativecommons.org/licenses/by-nc-sa/4.0/ ! You are free to Share & Adapt under the following terms: ! Give Credit, NonCommercial, ShareAlike ! ----- ! ! neue Counterwerte vom Zähler holen dom.GetObject("CUXD.CUX2801001:3.CMD_EXEC").State("tclsh /usr/local/addons/update_HB-ES-S0-CNT6.tcl"); </pre>				
sofort				
Skript				
<pre> ! ----- ! STROM.Update2.hms ! ! HomeMatic Skript für HB-ES-S0-CNT6 ! Version 0.22 ! 2019-09-03 Tom Major (Creative Commons) ! https://creativecommons.org/licenses/by-nc-sa/4.0/ ! You are free to Share & Adapt under the following terms: ! Give Credit, NonCommercial, ShareAlike ! ----- ! Die folgenden HomeMatic Systemvariablen müssen für diesen HomeMatic Skript angelegt werden: ! ! Typ Bereich Sichtbar Bemerkungen ! STROM.Cnt1 .. STROM.Cnt6 Zahl 0..2147000000 nein <intern> die Rohwerte der S0-Pulse (die vom TCL Skript upgedatet werden) ! STROM.LastCnt1 .. STROM.LastCnt6 Zahl 0..2147000000 nein <intern> die Rohwerte der S0-Pulse vom letzten Skript-Durchlauf ! STROM.Counter1 .. STROM.Counter6 Zahl 0..2147000000 ja aktuelle Stromzähler in kWh ! STROM.Power1 .. STROM.Power6 Zahl 0..2147000000 ja aktuelle Momentanverbräuche in Watt ! STROM.LastCalcTime Zahl 0..2147000000 nein <intern> der Zeitpunkt des letzten Skript-Durchlaufs ! STROM.ErrCnt Zahl 0..2147000000 ja Fehlerzähler des TCL Skripts ! STROM.ErrMsg String - ja letzte Fehlermeldung des TCL Skripts ! ! Falls CHANNEL_COUNT < 6 ist müssen natürlich auch nur so viele Systemvariablen pro Kategorie angelegt werden. ! Alle benötigten Systemvariablen können mit dem Skript STROM.CreateSysVars.hms automatisch erzeugt werden. ! ----- ! ! ===== ! START benutzerspezifische Konfiguration ! Kanalanzahl, Bereich 1..6 integer CHANNEL_COUNT = 5; ! Impulsparameter der S0-Zähler 1..6 in Imp/kWh string LIST_PULSE_KWH = "1000,2000,2000,2000,2000,2000"; ! ENDE benutzerspezifische Konfiguration ! ===== ! ! ----- ! Zeitdifferenz seit letztem Skript call ! falls es keine Unterbrechung gab sind dies 180 sec integer t0 = dom.GetObject("STROM.LastCalcTime").Value(); integer t1 = system.Date("%F %X").ToTime().ToInteger(); integer tDiff = t1 - t0; dom.GetObject("STROM.LastCalcTime").State(t1); ! ! bei ungewöhnlichen Differenzen abbrechen und beim nächsten Zyklus updaten (die HB-ES-S0-CNT6 Hardware behält ja ihren Zählerstand) ! andernfalls ist der Wert des Momentanverbrauch ggf. nicht plausibel if ((tDiff < 45) (tDiff > 1200)) { quit; } </pre>				

Name	Beschreibung	Bedingung (Wenn...)	Aktivität (Dann..., Sonst...)	Aktion
! -----				
! Deltas und Momentanverbrauch berechnen				
integer channel = 1;				
while (channel <= CHANNEL_COUNT) {				
! Namen der Systemvariablen vorbereiten				
string sysCur = "STROM.Cnt" # channel.ToString();				
string sysLast = "STROM.LastCnt" # channel.ToString();				
string sysPower = "STROM.Power" # channel.ToString();				
string sysCounter = "STROM.Counter" # channel.ToString();				
! Systemvariablen holen				
var curCnt = dom.GetObject(sysCur).Value();				
var lastCnt = dom.GetObject(sysLast).Value();				
! Deltaberechnungen				
if (curCnt >= lastCnt) {				
var pulseKWH = LIST_PULSE_KWH.StrValueByIndex(",", channel-1).ToInteger();				
if (pulseKWH == 0) {				
quit;				
}				
var counter = curCnt / pulseKWH;				
var diffCnt = curCnt - lastCnt;				
var power = 1000.0 * diffCnt / tDiff;				
dom.GetObject(sysLast).State(curCnt);				
dom.GetObject(sysCounter).State(counter.Round(2));				
dom.GetObject(sysPower).State(power);				
}				
channel = channel + 1;				
}				
verzögert um 15 Sekunden				
Aktivität: Sonst... <input type="checkbox"/> Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).				