

Name	Beschreibung	Bedingung (Wenn...)	Aktivität (Dann., Sonst..)	Aktion
Pelletsofen nach Raumtemp steuern 2	Nach Wohnzimmer Temp steuern	Kanalzustand: Wand-Wohnzimmer-Sensor bei Temperatur <i>im Wertebereich / mit Wert größer oder gleich 0.00° C</i> bei Aktualisierung auslösen	Skript: ... sofort ausführen	<input type="checkbox"/> systemintern
Bedingung: Wenn...				
Geräteauswahl <u>Wand-Wohnzimmer-Sensor</u> <i>bei Temperatur im Wertebereich / mit Wert größer oder gleich 0.00° C bei Aktualisierung auslösen</i>				
UND				
Systemzustand <u>Heizstatus-Pelletsofen</u> <i>bei Auto nur prüfen</i>				
UND				
Systemzustand <u>PelletsTempHigh</u> <i>im Wertebereich / mit Wert größer oder gleich 0.00 bei Aktualisierung auslösen</i>				
ODER				
Geräteauswahl <u>Wand-Wohnzimmer-Sensor</u> <i>bei Temperatur im Wertebereich / mit Wert größer oder gleich 0.00° C bei Aktualisierung auslösen</i>				
UND				
Systemzustand <u>Heizstatus-Pelletsofen</u> <i>bei Auto nur prüfen</i>				
UND				
Systemzustand <u>PelletsTempLow</u> <i>im Wertebereich / mit Wert größer oder gleich 0.00 bei Aktualisierung auslösen</i>				
Aktivität: Dann... <input checked="" type="checkbox"/> Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).				
Skript				
<pre>! Steuer den Pelletsofen anhand der Temperatur des Wandreglers im Wand-Wohnzimmer-TempSensor ! Verwendete Systemvariable ! PelletsTempHigh = Wird dieser Wert erreicht oder überschritten, wird der Aktor AUS geschaltet ! PelletsTempLow = Wird dieser Wert unterschritten, wird der Aktor EIN geschaltet ! Dieses Script sollte nur bei Temperaturprüfungen aufgerufen werden, wenn die Pelletsheizung aktiv ist !V0.01 Fehler im Namen des Temperatursensors behoben !Datenpunkt des Wohnzimmer Wandthermometer holen var myChannel = dom.GetObject("Wand-Wohnzimmer-Sensor"); var dp = myChannel.DPByHssDP("TEMPERATURE"); !Aktuelle Temperatur holen real istTemp = dp.Value(); !Datenpunkt des Schaltaktors holen myChannel = dom.GetObject("Pelletsofen-Aktor"); dp = myChannel.DPByHssDP("STATE"); ! Pruefen, ob Temperaturvorgabe HIGH erreicht oder ueberschritten wurde if (istTemp >= dom.GetObject("PelletsTempHigh").Variable()) { if (dp.Value() == true) { !Aktor abschaltbefehl, nur wenn Aktor auch eingeschaltet dp.State(0); ! Ofenstatus-Pellets auf StartCooldown, sofern Status nicht CoolDown oder StartCooldown ist if (dom.GetObject("Ofenstatus-Pellets").State() != 2) { if (dom.GetObject("Ofenstatus-Pellets").State() != 1) { dom.GetObject("Ofenstatus-Pellets").State(2); } } } } quit; } !Pruefen, ob Temperaturvorgabe LOW erreicht oder unterschritten wurde if (istTemp <= dom.GetObject("PelletsTempLow").Variable()) { ! Mindesttemperatur wurde unterschritten oder erreicht, Aktor nur einschalten wenn Status = Standby (0) if(dom.GetObject("Ofenstatus-Pellets").State() == 0) { if (dp.Value() == false) { !Aktor EIN Schalten, nur wenn ausgeschaltet dp.State(1); if (dom.GetObject("Ofenstatus-Pellets").State() != 3) { !Ofenstatus-Pellets auf Laeuft dom.GetObject("Ofenstatus-Pellets").State(3); } } } } quit; }</pre>				
sofort				
Aktivität:Sonst... <input type="checkbox"/> Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).				