

Heizungssteuerung Paradigma SystaComfort II in HomeMatic einbinden

Ab Software-Version 2.16 verfügen die Heizungssteuerungen Paradigma SystaComfort II und Compact C über eine Modbus-TCP - Schnittstelle. In diesem Beitrag zeige ich, wie sich die Daten der Heizungssteuerung an die Homematic (CCU3) übertragen lassen. Dazu verwende ich ioBroker, um die Systeme miteinander zu koppeln.

Durch die Kopplung von Heizungssteuerung und Homematic erhält die Homematic Zugriff auf die Datenpunkte des gesamten Heizungssystems. Dies beinhaltet die Messwerte des Systems, Status und Störmeldungen. Besonders interessant ist auch die Möglichkeit, Wärmemengen, z.B. von der Solaranlage abzufragen. Zusätzlich ist es auch möglich, dem Heizungsregler bestimmte Werte vorzugeben (Schreibzugriff). Allerdings sind hier die Möglichkeiten wie ich finde leider etwas sparsam.

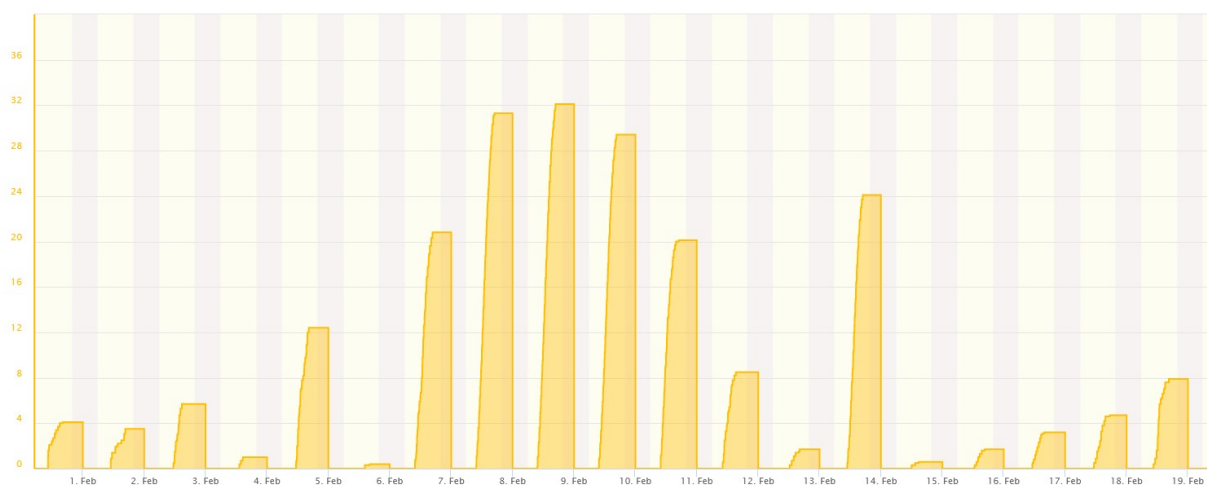
Voraussetzungen:

- Heizungssteuerung Paradigma SystaComfort II oder Compact C
- Software-Version der Steuerung V 2.16 oder höher
- Anschluss der Steuerung an das Hausnetzwerk (Ethernet – LAN)
- eine ioBroker-Installation

Als Grundlage für die Anbindung hat Paradigma ein ausführliches Dokument herausgegeben mit dem Titel „Modbus-Schnittstelle für Gebäudeleitsysteme“ (TH-3000). Es ist auf der Paradigma-Webseite www.paradigma.de/produkte/regelungen/systacomfortII/ verfügbar.

In dieser Anleitung möchte ich beschreiben, wie man die Homematic und die SystaComfort II über ioBroker koppeln und die Mess- und Statuswerte der SystaComfort II-Steuerung in Systemvariable der CCU3 abbilden kann.

Die Beeinflussung der Heizungssteuerung ist grundsätzlich auch möglich, wird in dieser Projektvorstellung aber nicht realisiert.



Beispiel: Messwert „Tagesenergie Solar (kWh)“ visualisiert über CCU-Historian HighChart

Schritt 1: SystaComfort II konfigurieren

Als erstes aktivieren wir den Modbus-TCP-Server der Heizungssteuerung. Dies ist nur am Hauptbedienteil der SystaComfort II möglich, also am physischen Bedienteil *S-Touch*. Der Menüpunkt für die Aktivierung ist nicht in der S-Touch – App oder evtl. installierten Heizkreis-Bedienteilen vorhanden.

Software-Version prüfen

Zunächst sollte unter `System > Kontrolle > Version` geprüft werden, ob die Software-Version der SystaComfort II die Modbus-Schnittstelle beinhaltet.

Die Version muss V 2.16 oder höher sein, ansonsten muss man ein Update der Software durch den Heizungstechniker durchführen lassen.



Software-Version prüfen

IP-Adresse SystaComfort II festlegen

Damit der ioBroker später mit der Heizungssteuerung kommunizieren kann, muss dieser eine feste IP-Adresse zugewiesen werden.

Dazu konfiguriert man am besten den DHCP-Server des Routers im Heimnetzwerk entsprechend. Alternativ kann man am Heizungsregler eine statische IP-Adresse eintragen. Die Vorgehensweise dafür ist in der Anleitung von Paradigma beschrieben.

Die zugewiesene IP-Adresse kann am Bedienteil unter `System > Netzwerk` geprüft werden. Gegebenenfalls muss der Heizungsregler nach Änderung der IP-Adresse neu gestartet werden, damit die Änderung übernommen wird.

Modbus-TCP – Server aktivieren

Jetzt kann am Hauptbedienteil der Modbus-TCP -Server der Heizungssteuerung im Menüpunkt `System > Einstellungen` aktiviert werden.

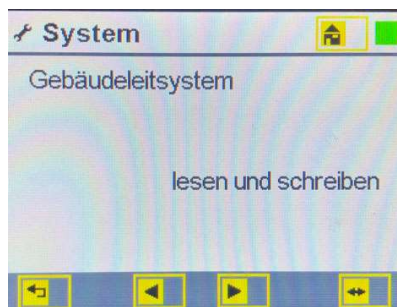
Hier gibt es folgende Einstellmöglichkeiten:

Menüpunkt	Werte	Werkseinstellung
Gebäudeleitsystem	aus	aus
	lesen	
	lesen und schreiben	
Port Gebäudeleitsystem	0 - 65535	502

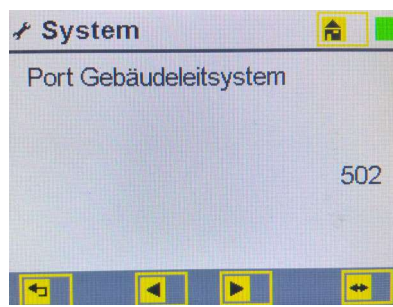
Für den Zugriff auf die SystaComfort muss die Einstellung `Gebäudeleitsystem` wenigstens auf `lesen` gestellt werden.

Möchte man auch Einstellungen der Steuerung über den ioBroker bzw. die Homematic verändern, muss hier `lesen und schreiben` gewählt werden.

Des Weiteren kann auch der Port des Modbus-Servers eingestellt werden. Ich verwende im Folgenden die Werkseinstellung Port 502.



Menüpunkt Gebäudeleitsystem



Menüpunkt Modbus-Port

Damit ist der Heizungsregler fertig konfiguriert und über Modbus-TCP aus dem Netzwerk erreichbar.

Dies lässt sich mit Hilfe eines Programms wie *QmodMaster* prüfen.

Eine detaillierte Anleitung dazu findet sich in Abschnitt 4 der Paradigma-Anleitung.

Dabei ist zu beachten, dass die Zählweise von Datenpunkten bei Modbus-Geräten unterschiedlich sein kann. Daher kann es sein, dass zur Adresse eines Datenpunkts noch +1 addiert werden muss.

Das beschreibt Paradigma auch ausführlich in Abschnitt 3, vergisst allerdings in der Beschreibung in Abschnitt 4, dass dies auch bei *QmodMaster* der Fall ist.

Damit das angegebene Beispiel mit Test-Register 9998 richtig funktioniert, musste ich in *QmodMaster* die Start-Adresse 9999 eingeben.

Hinweis:

Es kann immer nur ein Modbus-Slave mit der SysteComfort II verbunden sein. Wenn im Folgenden die Verbindung mit *ioBroker* hergestellt wird, muss *QmodMaster* von der Steuerung getrennt werden.

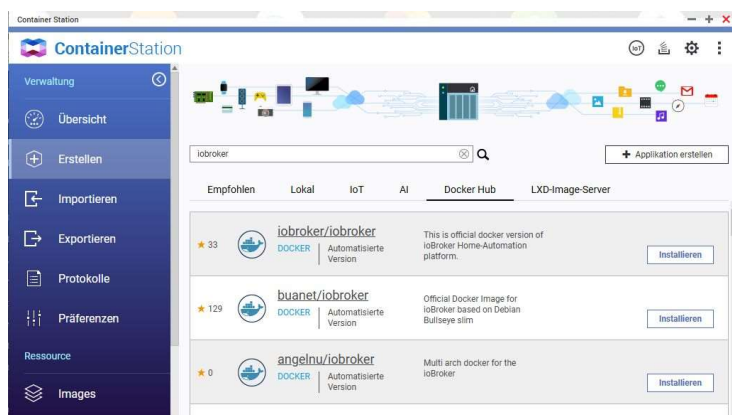
Schritt 2: ioBroker einrichten

Da es viele unterschiedliche Möglichkeiten gibt, einen ioBroker zu installieren, gehe ich im Folgenden davon aus, dass bereits eine ioBroker-Installation vorhanden ist.

Wichtig ist, dass der ioBroker dauerhaft läuft und eine feste IP-Adresse besitzt.

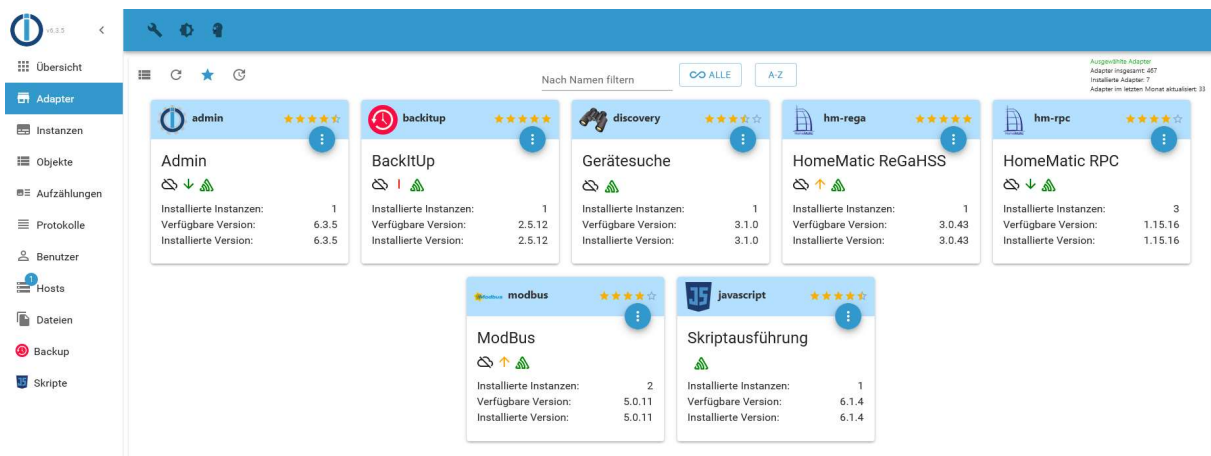
Bei mir läuft der ioBroker zum Beispiel in einem Docker-Container auf einem NAS. Dazu habe ich den bekannten *buynet/iobroker* – Container verwendet. Anleitungen zur Installation von ioBroker gibt es für verschiedene Szenarien im Web.

Tipp: Besonders bei der Erstellung eines Docker-Containers auf die Einstellung einer festen IP-Adresse achten.



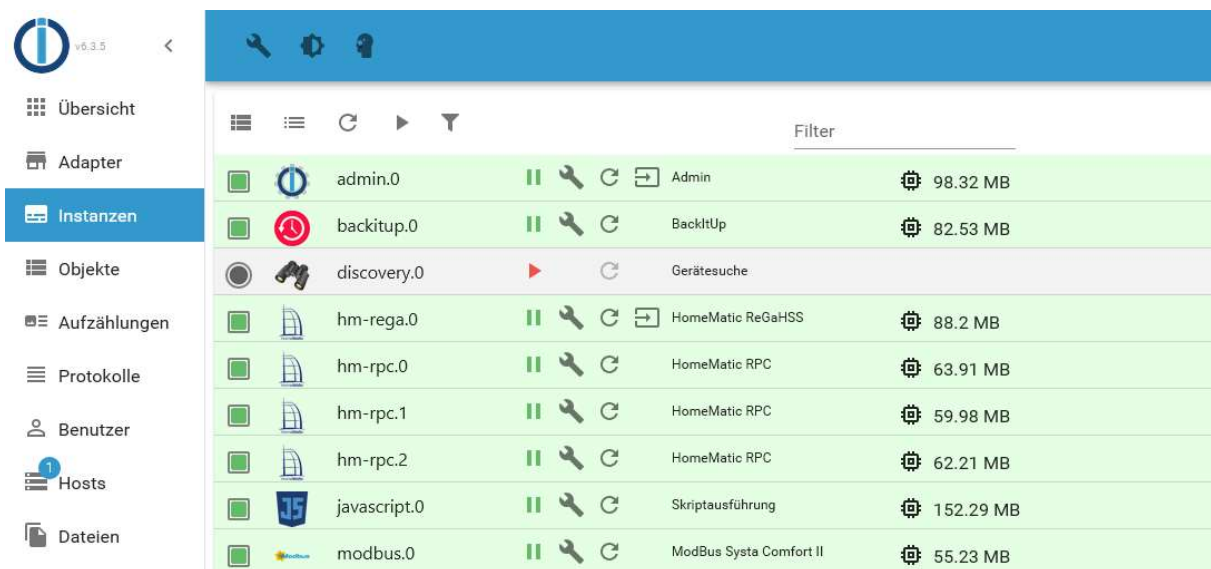
Beispiel: ioBroker als Docker-Container für QNAP-NAS installieren

Die Darstellungen in der weiteren Beschreibung beziehen sich auf ioBroker in der Version v6.3.5. Um Verbindung mit der SystaComfort II aufnehmen zu können, muss zunächst der ModBus-Adapter installiert werden. Dies erfolgt im ioBroker unter **Adapter**.



Soweit nicht bereits erfolgt, wird nun eine Instanz des ModBus-Adapters erstellt. Dazu klickt man auf dem ModBus-Adapter auf das Symbol und dann unten links auf das + („Instanz hinzufügen“).

Im Menü **Instanzen** sollte nun der Adapter *modbus.0* aufgeführt sein. Diesen können wir zur besseren Übersicht die Bezeichnung „ModBus Systa Comfort II“ geben:



Sollte bereits ein anderes Gerät über ModBus mit dem ioBroker verbunden sein, werden entsprechend mehrere ModBus-Instanzen in der Liste vorhanden sein.

Nun klicken wir auf das Schraubenschlüssel-Symbol, um die Konfiguration der ModBus-Instanz vorzunehmen.

Zunächst sind hier in der Registerkarte **Allgemein** einige Einträge vorzunehmen, um die Verbindung mit der SystaComfort II herzustellen:

Instanzeinstellungen: modbus.0

Verbindungsparameter
 TCP/Serielle RTU
 TCP
 Partner IP-Adresse: 192.168...
 Port: 502
 Geräte ID: 1
☐ Mehrere Geräte-IDs
 Typ: Master

Allgemein
☒ Aliases benutzen
☐ Direkte Adressen benutzen (bei Aliases)
☒ Die Adressen nicht auf 16 Bits ausrichten
☐ "Mehrere Register schreiben" nicht verwenden
Nur mit FC5/FC6 schreiben
☐ Nur "Write multiple registers" verwenden
Nur mit FC15/FC16 schreiben
 Zahlen runden auf: 2
 Datenabfrageintervall: 40000 ms
 Wartezeit bis zum erneuten verbinden: 60000 ms
 Wartezeit Lesend: 5000 ms
 Impulszeit: 1000 ms
 Wartezeit: 50 ms
 Max Leseanforderungslänge (Float): 100 Register
 Max Leseanforderungslänge (Booleans): 128 Register
 Leseintervall: 0 ms
 Schreibintervall: 0 ms
☐ Unveränderte Zustände aktualisieren
☒ Adresse nicht in ID aufnehmen
☐ Punkte in IDs erhalten

Unter **Verbindungsparameter** links ist die feste IP-Adresse anzugeben, die im Schritt 1 der SystaComfort II zugewiesen wurde sowie der Port, der in der Heizungssteuerung eingestellt ist (Default-Wert: 502).

Auf der rechten Seite werden die weiteren Verbindungseigenschaften eingestellt. Wichtig ist hier, den Haken bei **Die Adressen nicht auf 16 Bits ausrichten** zu setzen.

Paradigma gibt außerdem vor, dass die Wartezeit bei der zyklischen Aktualisierung von Datenpunkten mindestens 10 Sekunden betragen muss.

Ich habe ein **Datenabfrageintervall** von 40 000 ms = 40 sek. eingestellt.

Nun geht es an das Anlegen der Datenpunkte.

Die Paradigma-Steuerung verwendet drei Datenpunkt-Typen:


- Diskrete Ausgänge
- Eingangsregister
- Holding-Register

Alle Datenpunkte sind in Anhang A der Paradigma-Beschreibung aufgeführt, wobei beachtet werden muss, dass zu der Adresse jeweils +1 addiert werden muss.

(*Diskrete Eingänge* werden entsprechend der Nomenklatur von ModBus mit *Coils* bezeichnet.)

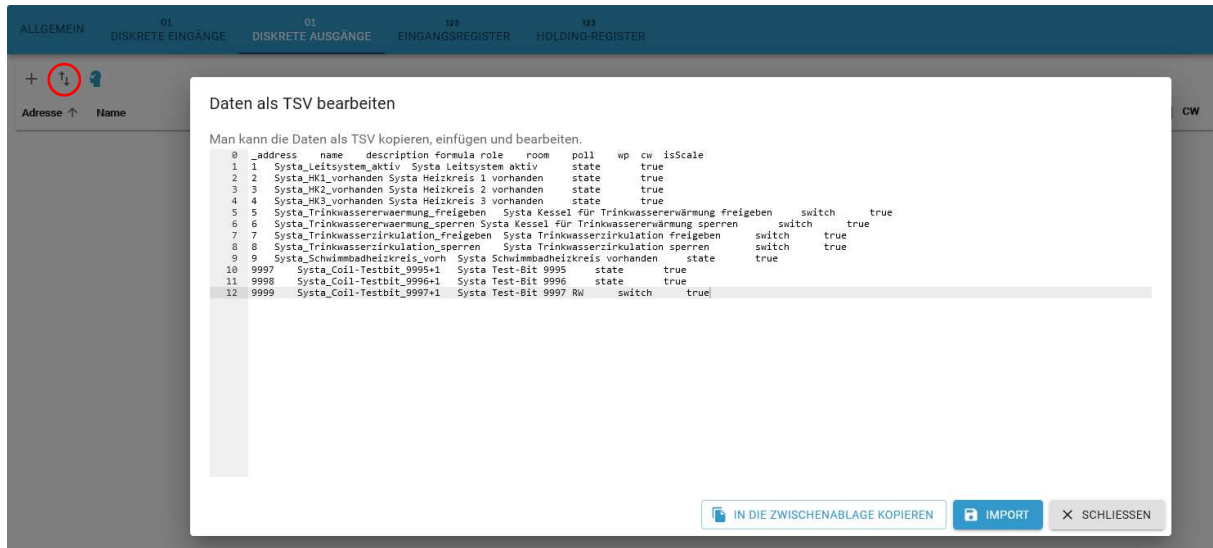
Zum einfachen Anlegen der Datenpunkte können die vorbereiteten TSV-Dateien (tabulatorgetrennte Werte) verwendet werden. Mit diesen Dateien können die Einstellungen direkt in die jeweiligen Datenpunkt-Typen importiert werden.

Hier am Beispiel für Diskrete Ausgänge:

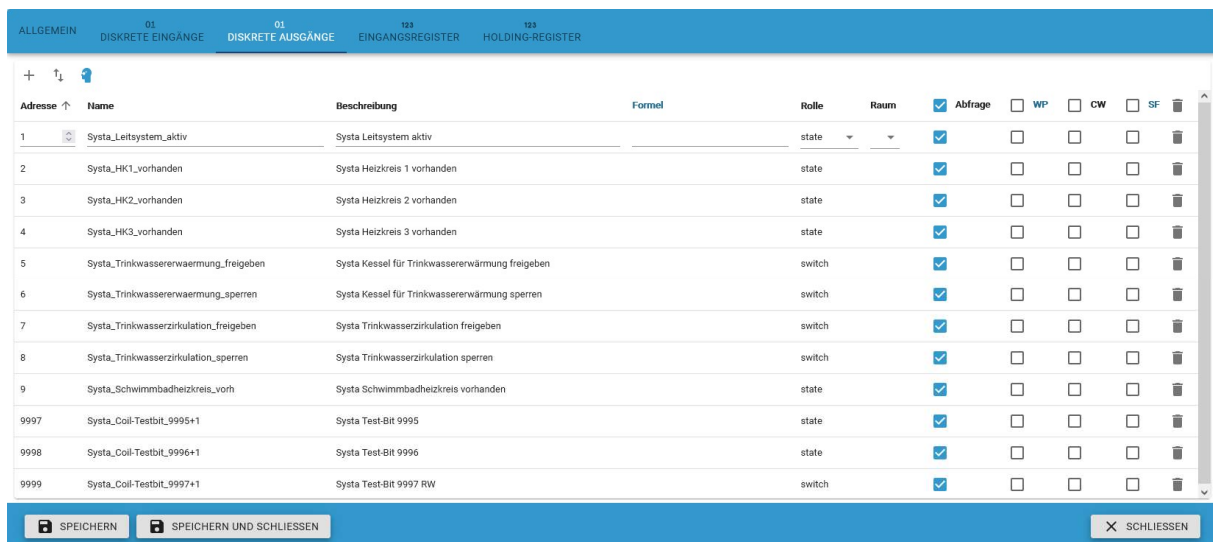
In die Registerkarte Diskrete Ausgänge wechseln und auf das Symbol  (Als TSV bearbeiten) klicken. Es öffnet sich der Dialog *Daten als TSV bearbeiten*.

Den vorhandenen Eintrag löschen, so dass die Liste ganz leer ist.

Nun mit Copy-and-Paste den Inhalt der Datei `Modbus_Systa2_Diskrete Ausgänge.txt` einfügen:



Nach Klicken auf Import sollte das Ergebnis folgendermaßen aussehen:



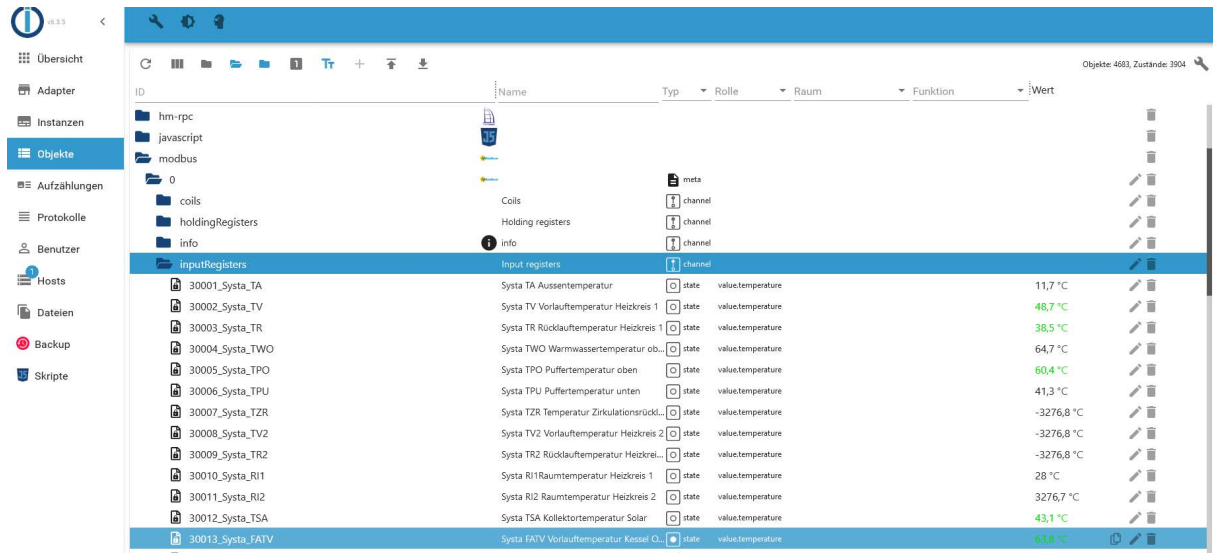
Speichern klicken, um die Datenpunkte zu sichern und den Vorgang entsprechend für Eingangsregister und Holding-Register wiederholen.

Damit sind alle vorhandenen Datenpunkte importiert und die ModBus-Instanz kann im Menüpunkt Instanzen gestartet werden.

Wenn alles geklappt hat, sollte der Status des Adapters etwa so aussehen:



Im Menü Objekten sollten nur im Ordner modbus > 0 die Datenpunkte erscheinen und mit Werten gefüllt werden:



Wie man sehen kann, werden nicht alle Datenpunkte mit sinnvollen Werten gefüllt.

Dies hängt mit der konkreten Anlagenkonfiguration zusammen. Im Screenshot sieht man z.B., dass der Datenpunkt Syste RI2 Raumtemperatur Heizkreis 2 auf 3276,7 °C steht. In meiner Anlage ist nur ein Heizkreis vorhanden, so dass hier kein sinnvoller Wert übertragen wird.

Es macht Sinn, sich die einzelnen Datenpunkte anzuschauen und ggf. unnütze Einträge zu löschen, um das System schlank zu halten.

Schritt 3: HomeMatic Systemvariablen vorbereiten

Die Datenpunkte aus dem ioBroker sollen in Systemvariable der HomeMatic CCU gespiegelt werden. Dazu legen wir in diesem Schritt die entsprechenden Variablen an.

Sinnvollerweise beschränkt man sich hier bereits auf die Variablen, die im eigenen System sinnvoll sind.

Um im ioBroker die Systemvariablen der HomeMatic später leicht von den Datenpunkten der Heizungssteuerung auseinanderhalten zu können, lasse ich die Namen der Systemvariablen mit „SysteHM“ beginnen.

Die folgende Tabelle zeigt als Beispiel die von mir benutzten Datenpunkte, für die ich entsprechend Systemvariable in der CCU angelegt habe. Die gewünschten Systemvariablen müssen in diesem Schritt in der CCU angelegt werden:

Name	Beschreibung	Variablentyp	Werte	Maßeinheit
SystaHM_Anzahl_Kesselstarts	Systa Anzahl Starts Kessel 1	Zahl	Minimalwert: 0 Maximalwert: 1000000	
SystaHM_Betriebsart_HK1	Systa Betriebsart Heizkreis 1	Zahl	Minimalwert: 0 Maximalwert: 100	
SystaHM_Betriebsart_Schwimmbad	Systa Betriebsart Schwimmbad	Zahl	Minimalwert: 0 Maximalwert: 100	
SystaHM_Betriebsstd_Kessel1	Systa Betriebsstunden Kessel 1	Zahl	Minimalwert: 0 Maximalwert: 10000000	h
SystaHM_FATR	Systa Rücklauftemp Kessel	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_FATV	Systa Vorlauftemp Kessel	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_Gesamtenergie_Solar	Systa Gesamtenergie Solarertrag	Zahl	Minimalwert: 0 Maximalwert: 999999999	kWh
SystaHM_Kesselsolltemp	Systa Kesselsolltemperatur	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_Kollektorleistung_Solar	Systa Kollektorleistung Solar	Zahl	Minimalwert: 0 Maximalwert: 1000	kW
SystaHM_Leitsystem_aktiv	Systa Leitsystem aktiv	Logikwert	wahr = aktiv falsch = inaktiv	
SystaHM_RI1	Systa Raumtemp HK 1	Zahl	Minimalwert: -100 Maximalwert: 200	°C
SystaHM_S	Systa Volumenstrom Solar	Zahl	Minimalwert: 0 Maximalwert: 100	l/min
SystaHM_Schwimmbadheizkreis_vorh	Systa Schwimmbadheizkreis vorhanden	Logikwert	wahr = vorhanden falsch = nicht vorhanden	
SystaHM_Solarertrag_letzte_Std	Systa Solarertrag letzte Stunde (berechnet)	Zahl	Minimalwert: -1 Maximalwert: 1000	kWh
SystaHM_Soll-Leistung_HK1	Systa Sollwert Leistung HK1	Zahl	Minimalwert: 0 Maximalwert: 100	%
SystaHM_Solltemp_Puffer_oben	Systa Solltemperatur Puffer oben	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_Sollwert_Trinkwassertemp	Systa Sollwert Trinkwassertemp (R/W)	Zahl	Minimalwert: 0 Maximalwert: 100	°C
SystaHM_Status_Heizkessel	Systa Status Heizkessel	Zahl	Minimalwert: 0 Maximalwert: 100	
SystaHM_Status_Heizkreis1	Systa Status Heizkreis 1	Zahl	Minimalwert: 0 Maximalwert: 100	
SystaHM_Status_Schwimmbad	Systa Status Schwimmbad	Zahl	Minimalwert: 0 Maximalwert: 100	
SystaHM_Status_Solar	Systa Status Solar	Zahl	Minimalwert: 0 Maximalwert: 100	
SystaHM_Status_Warmwasser	Systa Status Warmwasser	Zahl	Minimalwert: 0 Maximalwert: 100	
SystaHM_Status_Zirkulation	Systa Status Zirkulation	Zahl	Minimalwert: 0 Maximalwert: 100	
SystaHM_Stoercode_Heizungsregler	Systa Störcode Heizungsregler	Zahl	Minimalwert: 0 Maximalwert: 65535	
SystaHM_Stoercode_Kessel1	Systa Störcode Wärmeerzeuger 1	Zahl	Minimalwert: 0 Maximalwert: 65535	
SystaHM_Stoercode_Solar	Systa Störcode Solarregler	Zahl	Minimalwert: 0 Maximalwert: 65535	
SystaHM_TA	Systa Aussentemperatur	Zahl	Minimalwert: -100 Maximalwert: 100	°C

SystaHM_Tagesenergie_Solar	Systa Tagesenergie Solar	Zahl	Minimalwert: 0 Maximalwert: 10000	kWh
SystaHM_TAM	Systa Aussentemp am Kollektor	Zahl	Minimalwert: -100 Maximalwert: 200	°C
SystaHM_TKW	Eintrittstemp Kaltwasser Frischwasserstation	Zahl	Minimalwert: -100 Maximalwert: 200	°C
SystaHM_TPO	Systa Puffertemp oben	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_TPU	Systa Puffertemp unten	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_TR	Systa Rücklauftemp Heizkreis	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_Trinkwassererw_freigegeben	Systa Trinkwassererwärmung freigegeben (R/W)	Logikwert	wahr = freigegeben falsch = nicht freigegeben	
SystaHM_Trinkwassererw_sperren	Systa Trinkwassererwärmung sperren (R/W)	Logikwert	wahr = gesperrt falsch = nicht gesperrt	
SystaHM_Trinkwasserzirkulation_freigegeben	Systa Trinkwasserzirkulation freigegeben	Logikwert	wahr = freigegeben falsch = nicht freigegeben	
SystaHM_Trinkwasserzirkulation_sperren	Systa Trinkwasserzirkulation sperren	Logikwert	wahr = gesperrt falsch = nicht gesperrt	
SystaHM_TRSB	Systa Rücklauftemp HK Pool	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_TSA	Systa Kollektortemp. Solar	Zahl	Minimalwert: -100 Maximalwert: 1000	°C
SystaHM_TSB	Systa Temperatur Schwimmbad	Zahl	Minimalwert: -100 Maximalwert: 200	°C
SystaHM_TSBsoll	Systa Solltemp Schwimmbad bei Erwärmung durch Heizkessel (R/W)	Zahl	Minimalwert: 0 Maximalwert: 100	°C
SystaHM_TSBsoll_solar	Systa Solltemp Schwimmbad bei Heizen mit solarer Überschusswärme (R/W)	Zahl	Minimalwert: 0 Maximalwert: 100	°C
SystaHM_TSE	Systa Solare Eintrittstemp	Zahl	Minimalwert: -100 Maximalwert: 500	°C
SystaHM_TSP	Systa Eintrittstemp Speicher in Frischwasserstation	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_TSV	Systa Temp Solarvorlauf	Zahl	Minimalwert: -100 Maximalwert: 500	°C
SystaHM_TV	Systa Vorlauftemp Heizkreis	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_TV_soll	Systa Sollwert Vorlauftemp HK1 (R/W)	Zahl	Minimalwert: 0 Maximalwert: 100	°C
SystaHM_TVSB	Systa Vorlauftemp HK Pool	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_TW	Systa Temp Speicher oben	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_TWO	Systa Warmwassertemp oben	Zahl	Minimalwert: 0 Maximalwert: 250	°C
SystaHM_TWW	Systa Austrittstemp Warmwasser	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_TZR	Systa Temp. Zirkulationsrücklauf	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_VKW	Systa Volumenstrom Kaltwasser	Zahl	Minimalwert: 0 Maximalwert: 100	l/min
SystaHM_Vorlauftemp_max_HK1	Systa max. Vorlauftemp HK1	Zahl	Minimalwert: 0 Maximalwert: 200	°C
SystaHM_VSPm	Systa Volumenstrom Speicherseite	Zahl	Minimalwert: 0 Maximalwert: 100	l/min

SystaHM_Waermemenge_Warmwasser	Systa Wärmemenge Warmwasser	Zahl	Minimalwert: 0	kWh
			Maximalwert: 10000000	
SystaHM_Waermemenge_Zirkulation	Systa Wärmemenge Zirkulation	Zahl	Minimalwert: 0	kWh
			Maximalwert: 10000000	

Hinweis:

Die Variable *SystaHM_Solarertrag_letzte_Std* ist kein Datenpunkt der SystaComfort II, sondern wird im ioBroker berechnet.

Schritt 4: Verbindung HomeMatic zu ioBroker herstellen

Damit die im ioBroker vorhandenen Datenpunkte der SystaComfort II in die Systemvariablen der CCU übertragen werden können, muss eine Verbindung zwischen HomeMatic und ioBroker hergestellt werden.

Zu diesem Thema gibt es schon viele gute Anleitungen im Web, so dass ich das hier nur ganz kurz beschreiben möchte.

Eine Anleitung bei YouTube, die mir gut geholfen hat ist z.B. „Homematic CCU mit ioBroker Verbinden“ von MK-IT.

Im ioBroker gibt es für die HomeMatic grundsätzlich zwei Adaptertypen:

- *HomeMatic ReGaHSS* verbindet die Logikschicht der CCU mit ioBroker. Dieser Adapter wird (u.a.) für den Austausch der Systemvariablen gebraucht.
- *HomeMatic RPC* verbindet die Interface-Prozesse der CCU mit dem ioBroker. Darunter fallen vereinfacht gesagt die Datenpunkte der Geräte (BidCos etc.) und Dinge wie der CUxD.

Wir benötigen hier eigentlich nur den *HomeMatic ReGaHSS* – Adapter. Es kann aber gegebenenfalls Sinn machen, bei der Gelegenheit auch den *HomeMatic RPC* - Adapter mitzuinstallieren, um ein komplettes Abbild der Homematic im ioBroker zu haben.

Beim *HomeMatic RPC* – Adapter ist zu beachten, dass in der Regel mehrere Instanzen angelegt werden müssen für die verschiedenen Dienste in der HomeMatic. Beispielsweise einer für drahtlose Geräte (rfd), HomeMatic wired, CUxD, HomeMatic IP usw., je nachdem, was im System vorhanden ist.

Bei der Einrichtung sollte mit den *HomeMatic RPC* – Instanzen begonnen werden.

Hier wird jeweils die IP-Adresse der HomeMatic eingetragen und unter *Adapter Adresse* die IP-Adresse des ioBrokers.


Bei *Daemon* wird entsprechend des jeweiligen Dienstes ausgewählt, ob diese Instanz rfd, CUxD usw. unterstützen soll (für jeden Dienst muss eine eigene Instanz des Adapters erstellt werden). Die Einstellung *HomeMatic Port* wird bei Auswahl des Dienstes automatisch angepasst und muss in der Regel nicht geändert werden.

Im Anschluss wird die Instanz des *HomeMatic ReGaHSS* – Adapters konfiguriert.

Unter *Haupteinstellungen* muss zunächst wieder die IP-Adresse der HomeMatic CCU eingetragen werden:

Instanzeinstellungen: hm-rega.0
v3.0.43

HAUPTEINSTELLUNGEN
SYNCHRONISIERE



HomeMatic CCU-Adresse
192.168.

Wiederverbindungsintervall (s)
30

rfd
☒ Aktiviert

hm-rpc-Instanz
hm-rpc.0

hs485d (Wired)
☐ Aktiviert

hm-rpc-Instanz
keins

CUxD
☒ Aktiviert

hm-rpc-Instanz
hm-rpc.2

HomeMatic IP
☒ Aktiviert

hm-rpc-Instanz
hm-rpc.1

Virtual Devices
☐ Aktiviert

hm-rpc-Instanz
keins

zyklisches Abfragen
☒ Aktiviert

Intervalle (s)
30

Trigger
BidCoS-RF.50.PRESS_SHORT

Dann werden die gewünschten Dienste angehakt, für die man zuvor die entsprechenden hm-rpc-Instanzen angelegt hat. Mit den Drop-Down – Listen rechts wird die (richtige!) Instanz jeweils dem angehakten Dienst zugeordnet.

Nun wechseln wir auf den Reiter *Synchronisiere* und haken die Elemente an, die mit dem ioBroker synchronisiert werden sollen:

Instanzeinstellungen: hm-rega.0
v3.0.43
HAUPT-EINSTELLUNGEN
SYNCHRONISIERE

DutyCycle	<input type="checkbox"/> Aktiviert	Intervalle (s)
Variablen	<input checked="" type="checkbox"/> Aktiviert	<input type="checkbox"/> Unsichtbare Variablen
Programme	<input checked="" type="checkbox"/> Aktiviert	
Namen	<input checked="" type="checkbox"/> Aktiviert	
Favoriten	<input checked="" type="checkbox"/> Aktiviert	speichere in der Aufzählung enum.favorites
Räume	<input checked="" type="checkbox"/> Aktiviert	speichere in der Aufzählung enum.rooms
Gewerke	<input checked="" type="checkbox"/> Aktiviert	speichere in der Aufzählung enum.functions

Bevor der ioBroker auf die CCU zugreifen kann, müssen dort noch ein paar Einstellungen vorgenommen werden.

Zunächst muss unter `Einstellungen > Systemsteuerung > Zusatzsoftware` die *XML-API* installiert werden.

Dann müssen wir noch die Einstellung der Firewall der CCU anpassen (`Einstellungen > Systemsteuerung > Firewall konfigurieren`).

Hier muss eine Freigabe für die XML-RPC API eingestellt werden.
Meine Empfehlung ist, die Freigaben nur auf das Notwendige zu beschränken.

Dazu die Einstellung `Homematic XML-RPC API auf Eingeschränkt` stellen.

Nun müssen die benötigten Ports im Feld `Port-Freigabe` mit Semikolon getrennt eingetragen werden. Die benötigten Ports können im ioBroker für jede der angelegten Instanzen der HomeMatic RPC abgelesen werden (`Haupteinstellungen > HomeMatic-Port`).

In der Liste IP-Adressen für den eingeschränkten Zugriff wird nun noch die IP-Adresse des ioBrokers hinzugefügt.

Hinweis:

Wenn der ioBroker in einem Docker-Container ausgeführt wird und keine Verbindung zur HomeMatic CCU hergestellt werden kann, sollten die Port-Einstellungen des Containers geprüft werden. Die entsprechenden Ports müssen freigegeben und 1:1 durchgereicht werden.

CCU - Firewall

Firewall-Richtlinie:
Ports blockiert

Zugriffseinstellungen der Ports

Homematic XML-RPC API:
Eingeschränkt

Ermöglicht den direkten Zugriff auf angelernte Homematic Geräte

Remote Homematic-Script API:
Eingeschränkt

Ermöglicht den Zugriff auf die Logikschicht der Homematic Zentrale

Mediola-Zugriff:
Eingeschränkt

Ermöglicht den Zugriff auf den Mediola-Service

Port-Freigabe:

8701;
2001;
2010

Hier können benötigte Ports freigegeben werden. Geben Sie die Ports durch ein ';' getrennt ein.

IP-Adressen für den eingeschränkten Zugriff:

192.168
192.168
fc00::/7
192.168
192.168

Sie können den Zugriff für einzelne IP-Adressen oder ganze Adressbereiche sowohl für IPv4 als auch für IPv6 freigeben. Geben Sie die Adressen durch ein ';' getrennt ein.

Abbrechen

OK

Konfiguration der CCU-Firewall

Schließlich können im ioBroker unter Instanzen die verschiedenen HomeMatic – Adapter gestartet werden und sollten alle grün werden:

<div>Instanzen</div> <div>Objekte</div> <div>Aufzählungen</div> <div>Protokolle</div> <div>Benutzer</div> <div>Hosts</div> <div>Dateien</div> <div>Backup</div> <div>Skripte</div>					
		backup.0		BackitUp	81.24 MB
		discovery.0		Gerätesuche	
		hm-rega.0		HomeMatic ReGaHSS	88.68 MB
	hm-rega.0 v3.0.43 <div> <div>✓</div> Verbunden mit Host <div>✓</div> Lebenszeichen <div>✓</div> Verbunden mit Gerät oder Dienst </div>				
		hm-rpc.0		HomeMatic RPC	64.41 MB
		hm-rpc.1		HomeMatic RPC	59.08 MB
		hm-rpc.2		HomeMatic RPC	63.97 MB
		javascript.0		Skriptausführung	149.42 MB
		modbus.0		ModBus Sysa Comfort II	55.4 MB

Im ioBroker sollte sich nun unter `Objekte` nach und nach die Datenpunkte, Objekte und Programme aus der CCU einfinden.
Die Systemvariablen befinden sich im Ordner `hm-rega > 0`, wobei dort zuerst alle Programme (mit Ordnersymbol) aufgelistet sind und danach die Systemvariablen.

Schritt 5: Synchronisierung der Datenpunkte vorbereiten

Bis jetzt haben wir es geschafft, im ioBroker die Datenpunkte der beiden Geräte – SystaComfort II und HomeMatic CCU – abzubilden.

Nun müssen wir noch dafür sorgen, dass die Werte der Systemvariablen der CCU mit den Datenpunkten der SystaComfort II beschrieben werden. Also dass bei jeder Werteänderung in der Heizungssteuerung die entsprechende Systemvariable aktualisiert wird.

Für diese Aufgabe verwende ich ein Script in ioBroker, das ich mit *Blockly* erstellt habe.

Als Vorbereitung dafür muss zuerst der Adapter *Skriptausführung (javascript)* installiert werden.

Nach der Installation erscheint im Menü von ioBroker ein neuer Punkt *Skripte*.

Wir wechseln in diesen Menüpunkt und erstellen einen neuen Ordner, den ich z.B.

„*ParadigmaSystaComfort2*“ genannt habe.

In diesem Ordner erstellen wir ein neues Skript, das ich „*Systa2Homematic*“ genannt habe.

Nun kommt leider etwas Handarbeit.

Für jeden Datenpunkt, der von der Heizungssteuerung in eine Systemvariable synchronisiert werden soll, benötigen wir eine Programmanweisung.

Blockly enthält für diesen Zweck eine sehr praktische Konstruktion mit dem Namen „*binde object... mit...*“ (unter „*System*“):



Dies bewirkt, dass das `Object ID 2` immer den gleichen Wert erhält wie `Object ID 1`, ohne dass eine weitere Programmkonstruktion notwendig wäre.

Mit der angehakten Option *nur Änderungen* wird `Object ID 2` nur bei einer Werteänderung von `Object ID 1` aktualisiert, nicht jedoch, wenn `Object ID 1` lediglich mit dem gleichen Wert aktualisiert wurde.

Für die oben aufgeführte Liste mit Systemvariablen sieht das Blockly-Programm dann so aus:

Coil Registers Systa Comfort II nach Homematic

binde object	Objekt ID Systa Leitsystem aktiv	mit	Objekt ID SystaHM_Leitsystem_aktiv	nur Änderungen✓
binde object	Objekt ID Systa Schwimmbadheizkreis vorhanden	mit	Objekt ID SystaHM_Schwimmbadheizkreis_vorh	nur Änderungen✓

Input Registers Systa Comfort II nach Homematic

binde object	Objekt ID Systa TA Aussentemperatur	mit	Objekt ID SystaHM_TA	nur Änderungen✓
binde object	Objekt ID Systa TV Vorlauftemperatur Heizkreis 1	mit	Objekt ID SystaHM_TV	nur Änderungen✓
binde object	Objekt ID Systa TR Rücklauftemperatur Heizkreis 1	mit	Objekt ID SystaHM_TR	nur Änderungen✓
binde object	Objekt ID Systa TWO Warmwassertemperatur oben	mit	Objekt ID SystaHM_TWO	nur Änderungen✓
binde object	Objekt ID Systa TPO Puffertemperatur oben	mit	Objekt ID SystaHM_TPO	nur Änderungen✓
binde object	Objekt ID Systa TPU Puffertemperatur unten	mit	Objekt ID SystaHM_TPU	nur Änderungen✓
binde object	Objekt ID Systa TZR Temperatur Zirkulationsrücklauf	mit	Objekt ID SystaHM_TZR	nur Änderungen✓
binde object	Objekt ID Systa RI1Raumtemperatur Heizkreis 1	mit	Objekt ID SystaHM_RI1	nur Änderungen✓
binde object	Objekt ID Systa TSA Kollektortemperatur Solar	mit	Objekt ID SystaHM_TSA	nur Änderungen✓
binde object	Objekt ID Systa FATV Vorlauftemperatur Kessel OTLAN	mit	Objekt ID SystaHM_FATV	nur Änderungen✓
binde object	Objekt ID Systa FATR Rücklauftemperatur Kessel OTLAN	mit	Objekt ID SystaHM_FATR	nur Änderungen✓
binde object	Objekt ID Systa TSB Schwimmbad-Temperatur ist	mit	Objekt ID SystaHM_TSB	nur Änderungen✓
binde object	Objekt ID Systa TVSB Vorlauftemperatur Schwimmbad-Heizkreis	mit	Objekt ID SystaHM_TVSB	nur Änderungen✓
binde object	Objekt ID Systa TRSB Rücklauftemperatur Schwimmbad-Heizkreis	mit	Objekt ID SystaHM_TRSB	nur Änderungen✓
binde object	Objekt ID Systa TSE Solare Eintrittstemperatur	mit	Objekt ID SystaHM_TSE	nur Änderungen✓
binde object	Objekt ID Systa TW Temperatur Speicher oben	mit	Objekt ID SystaHM_TW	nur Änderungen✓
binde object	Objekt ID Systa TSV Temperatur Solarverlauf	mit	Objekt ID SystaHM_TSV	nur Änderungen✓
binde object	Objekt ID Systa S Volumenstrom Solar	mit	Objekt ID SystaHM_S	nur Änderungen✓
binde object	Objekt ID Systa TAM Außentemperatur am Kollektor 1	mit	Objekt ID SystaHM_TAM	nur Änderungen✓
binde object	Objekt ID Systa TSP Eintrittstemperatur Speicher in die Fr...	mit	Objekt ID SystaHM_TSP	nur Änderungen✓
binde object	Objekt ID Systa TWW Austrittstemperatur Warmwasser Frischw...	mit	Objekt ID SystaHM_TWW	nur Änderungen✓
binde object	Objekt ID Systa TKW Eintrittstemperatur Kaltwasser Frischw...	mit	Objekt ID SystaHM_TKW	nur Änderungen✓
binde object	Objekt ID Systa VKW Volumenstrom Kaltwasser	mit	Objekt ID SystaHM_VKW	nur Änderungen✓
binde object	Objekt ID Systa VSPm Volumenstrom Speicherseite	mit	Objekt ID SystaHM_VSPm	nur Änderungen✓

(Fortsetzung nächste Seite)

(Fortsetzung)

RO Holding Registers Systa Comfort II nach Homem...

binde object	Objekt ID Systa Sollwert Leistung Heizkreis 1	mit	Objekt ID SystaHM_Soll-Leistung_HK1	nur Änderungen✓
binde object	Objekt ID Systa max Vorlauftemperatur Heizkreis 1	mit	Objekt ID SystaHM_Vorlauftemp_max_HK1	nur Änderungen✓
binde object	Objekt ID Systa Störcode Heizungsregler	mit	Objekt ID SystaHM_Stoercode_Heizungsregler	nur Änderungen✓
binde object	Objekt ID Systa Störcode Solarregler	mit	Objekt ID SystaHM_Stoercode_Solar	nur Änderungen✓
binde object	Objekt ID Systa Störung Wärmeerzeuger 1	mit	Objekt ID SystaHM_Stoercode_Kessel1	nur Änderungen✓
binde object	Objekt ID Systa Kollektorleistung Solar	mit	Objekt ID SystaHM_Kollektorleistung_Solar	nur Änderungen✓
binde object	Objekt ID Systa Tagesenergie Solar	mit	Objekt ID SystaHM_Tagesenergie_Solar	nur Änderungen✓
binde object	Objekt ID Systa Gesamtenergie Solar	mit	Objekt ID SystaHM_Gesamtenergie_Solar	nur Änderungen✓
binde object	Objekt ID Systa Wärmemenge Warmwasser	mit	Objekt ID SystaHM_Waermemenge_Warmwasser	nur Änderungen✓
binde object	Objekt ID Systa Wärmemenge Zirkulation	mit	Objekt ID SystaHM_Waermemenge_Zirkulation	nur Änderungen✓
binde object	Objekt ID Systa Betriebsstunden Kessel 1	mit	Objekt ID SystaHM_Betriebsstd_Kessel1	nur Änderungen✓
binde object	Objekt ID Systa Anzahl Starts Kessel 1	mit	Objekt ID SystaHM_Anzahl_Kesselstarts	nur Änderungen✓
binde object	Objekt ID Systa Status Warmwasser	mit	Objekt ID SystaHM_Status_Warmwasser	nur Änderungen✓
binde object	Objekt ID Systa Status Zirkulation	mit	Objekt ID SystaHM_Status_Zirkulation	nur Änderungen✓
binde object	Objekt ID Systa Status Heizkreis 1	mit	Objekt ID SystaHM_Status_Heizkreis1	nur Änderungen✓
binde object	Objekt ID Systa Status Solar	mit	Objekt ID SystaHM_Status_Solar	nur Änderungen✓
binde object	Objekt ID Systa Status Schwimmbad	mit	Objekt ID SystaHM_Status_Schwimmbad	nur Änderungen✓
binde object	Objekt ID Systa Status Heizkessel	mit	Objekt ID SystaHM_Status_Heizkessel	nur Änderungen✓
binde object	Objekt ID Systa Solltemperatur Puffer oben	mit	Objekt ID SystaHM_Solltemp_Puffer_oben	nur Änderungen✓
binde object	Objekt ID Systa Kesselsolltemperatur	mit	Objekt ID SystaHM_Kesselsolltemp	nur Änderungen✓
binde object	Objekt ID Systa Betriebsart Heizkreis 1	mit	Objekt ID SystaHM_Betriebsart_HK1	nur Änderungen✓
binde object	Objekt ID Systa Betriebsart Schwimmbad	mit	Objekt ID SystaHM_Betriebsart_Schwimmbad	nur Änderungen✓

RW Holding Register an CCU senden

binde object	Objekt ID Systa Sollwert Vorlauftemperatur Heizkreis 1	mit	Objekt ID SystaHM_TV_soll	nur Änderungen✓
binde object	Objekt ID Systa Sollwert Trinkwassertemperatur	mit	Objekt ID SystaHM_Sollwert_Trinkwassertemp	nur Änderungen✓
binde object	Objekt ID Systa Solltemperatur Schwimmbad TSBsoll	mit	Objekt ID SystaHM_TSBsoll	nur Änderungen✓
binde object	Objekt ID Systa Solltemperatur Schwimmbad TSBsoll_solar	mit	Objekt ID SystaHM_TSBsoll_solar	nur Änderungen✓

RW Coil Register an CCU senden

binde object	Objekt ID Systa Kessel für Trinkwassererwärmung freigeben	mit	Objekt ID SystaHM_Trinkwassererw_freigeben	nur Änderungen✓
binde object	Objekt ID Systa Kessel für Trinkwassererwärmung sperren	mit	Objekt ID SystaHM_Trinkwassererw_sperren	nur Änderungen✓
binde object	Objekt ID Systa Trinkwasserzirkulation freigeben	mit	Objekt ID SystaHM_Trinkwasserzirkulation_freigeben	nur Änderungen✓
binde object	Objekt ID Systa Trinkwasserzirkulation sperren	mit	Objekt ID SystaHM_Trinkwasserzirkulation_sperren	nur Änderungen✓

Die beiden letzten, abgesetzten Blöcke beinhalten die Bindungen derjenigen Datenpunkte, die nicht nur gelesen, sondern auch geschrieben werden können.

Dies ist momentan nicht von Bedeutung, weil hier nur die lesende Datenrichtung verwendet wird.

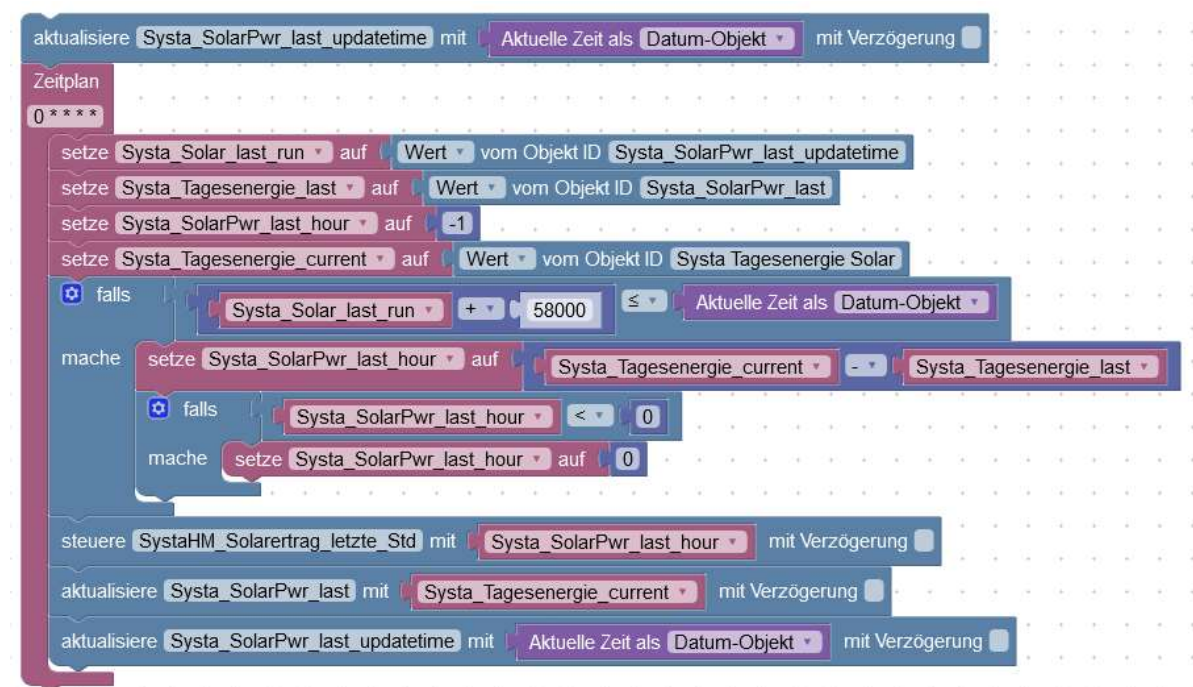
Soll die Schreibrichtung ergänzt werden, so muss ggf. bei diesen Datenpunkten eine andere Konstruktion erfolgen, um beide Richtungen implementieren zu können.

Nach dem Start des Skripts sollten nach und nach die Systemvariablen der HomeMatic mit den Werten der Heizungssteuerung aktualisiert werden.

Da die Aktualisierung nur dann erfolgt, wenn sich ein Wert ändert, kann das für einzelne Variablen ggf. recht lange dauern. Bei Temperaturwerten wie z.B. der Außentemperatur sollte relativ schnell ein Wert in der Systemvariablen auftauchen.

Zugabe: Solarertrag in der letzten Stunde berechnen

Mit den Werten der Heizungssteuerung lassen sich eine Reihe schöner Berechnungen anstellen. Einen Wert, den ich beispielsweise interessant fand, ist der Solarertrag der jeweils vergangenen Stunde. Damit lässt sich die Solarleistung über den Tag verteilt darstellen. Dieser Wert wird nicht direkt von der SystaComfort II geliefert, lässt sich aber leicht mit einem kleinen Programm erzeugen:



Programm „Solarertrag_berechnen“

Das Programm startet jede Stunde und bildet jeweils die Differenz des Werts des Datenpunkts `Systa Tagesenergie Solar` mit dem der vorangegangenen Stunde.

Diese wird in die Systemvariable `SystaHM_Solarertrag_letzte_Std` geschrieben.

Der Rest des Programms fügt ein paar Ausnahmebehandlungen hinzu.

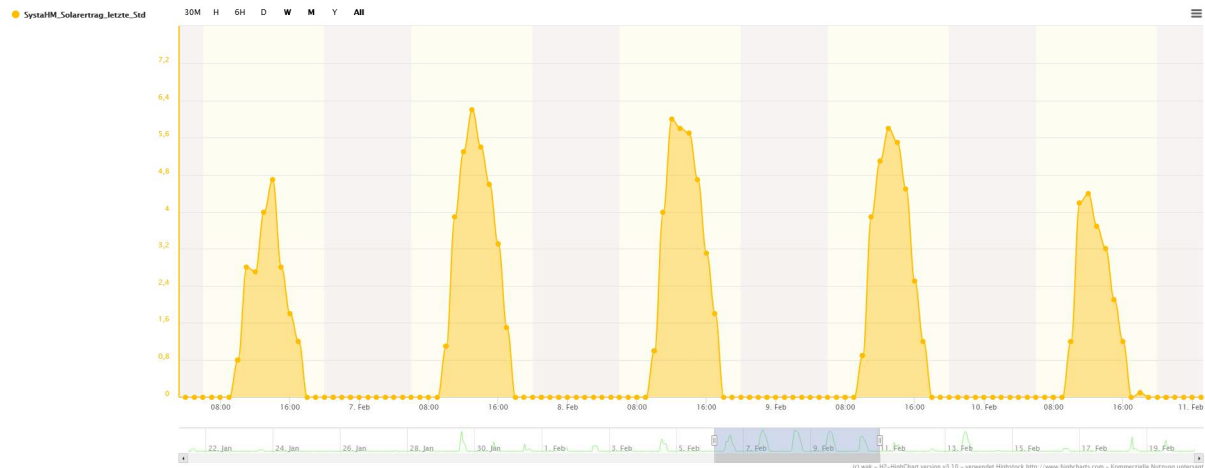
Im oberen Teil wird mit Hilfe der Variablen `Systa_SolarPwr_last_uptime` geprüft, ob der vergangene Zeitraum zwischen zwei Programmläufen genau eine Stunde beträgt oder ob dies aus irgendeinem Grund (Server down etc.) verpasst wurde. In diesem Fall wird -1 in `SystaHM_Solarertrag_letzte_Std` geschrieben.

Die zweite Ausnahmebehandlung tritt am Tagesende ein, wenn `Systa Tagesenergie Solar` von der Heizungssteuerung auf 0 gesetzt wird und das Ergebnis negativ wird.

Für das Programm benötigen wir zwei Variablen, deren Inhalt zwischen zwei Programmaufrufen nicht verloren geht:

- `Systa_SolarPwr_last` merkt sich den Wert des Solarertrags der letzten Stunde
- `Systa_SolarPwr_last_updatetime` speichert den Zeitpunkt des letzten Programmlaufs, um die Konsistenz des Stunden-Rhythmus' zu prüfen.

Damit diese Variablen dauerhaft gespeichert bleiben, legen wir sie als Objekte im Bereich `Objekte > 0_userdata > 0` an. Im Unterschied zu den lokalen Variablen, die nach dem Ende des Programms verloren gehen, werden diese Objekte über *Wert von Objekt ID ...* referenziert.



Ergebnis: Darstellung des Solarertrags pro Stunde (kWh/h = kW) im CCU-Historian

20.02.2023 - A0