

Migration von einer Homematic CCU2 zu IP Symcon 4.0

Ich möchte hier meinen Umstieg von meinem Aktuell rein mit einer Homematic CCU2 gesteuerten Smarthome auf ein IP Symcon gesteuertes System Dokumentieren das mir in Zukunft folgende Vorteile bieten wird:

- eine Bessere Integration von Anderen Systemen (1-Wire, KNX, usw...)
- Höhere Zuverlässigkeit (Timer ohne cuxd.timer in der CCU sind nicht zuverlässig!)
- Modernes Web UI (über den Reverse Proxy Nutzbar!)
- Möglichkeit der Integration einer Erweiterten Visualisierung.
- Einfache Skalierbarkeit der Zentralen Steuer Hardware (von einem PI bis hin zu einem x86 Server)

Warum das den nun?

Ich wollte weg von einer Reinen Bastel- Lösung wie es alle Alternativen aktuell sind und möchte auch in Zukunft Updates und auch Support.



Ziel ist es nur noch einmal alle Programme in ein anderes System zu migrieren sowie einen fließenden Übergang von CCU auf IP Symcon!

Nachdem die Version 4.0 von IP Symcon nun auch native unter Linux Lauffähig ist habe ich diese getestet und bin gerade dabei mein Homematic System auf IP Symcon zu migrieren.

Weiters gefällt mir PHP als Programmiersprache viel besser als die Alternativen von (FHEM, OpenHab, io.Broker usw...)

Konzept

In Zukunft wird es auch weiterhin eine CCU geben, diese wird aber nur mehr zum Anlernen von HM Komponenten und zum Verwalten der Direktverknüpfungen sowie von IP Symcon als „Gateway“ zu allen HM Funk und Wired Komponenten genutzt.

Hardware Anforderungen

Da ich meine Aktuelle CCU2 schon als LXCCU auf einem Raspberry PI 2 betreibe habe ich IP Symcon 4 direkt auf dem PI Installiert somit gibt es keinen bedarf einer neuen Hardware. Wenn natürlich in Zukunft die IP Symcon Installation wächst und die Leistung des PI nicht mehr ausreichen sollte kann ich die Installation einfach auf eine andere Leistungsfähigere Hardware migrieren.

Installation

IP Symcon am Server

Diese ist sehr einfach da nur das Repository und der Key von IP Symcon hinzugefügt werden muss und von da an bekommt ihr die Updates über das Betriebssystem.

Das Repository in die sources.list eintragen

```
sudo nano /etc/apt/sources.list
```

dort dann die Zeile

```
deb http://apt.ip-symcon.de/ stable main
```

den key installieren mit:

```
wget -q0 - http://apt.ip-symcon.de/symcon.key | sudo apt-key add -
```

und nun alles neu einlesen / aktualisieren mit:

```
sudo apt-get update && sudo apt-get upgrade
```

sowie ip symcon installieren:

```
sudo apt-get install symcon
```

Beim Installieren wird ein Dienst der beim neustart symcon automatisch startet angelegt dieser kann nun einfach mit:

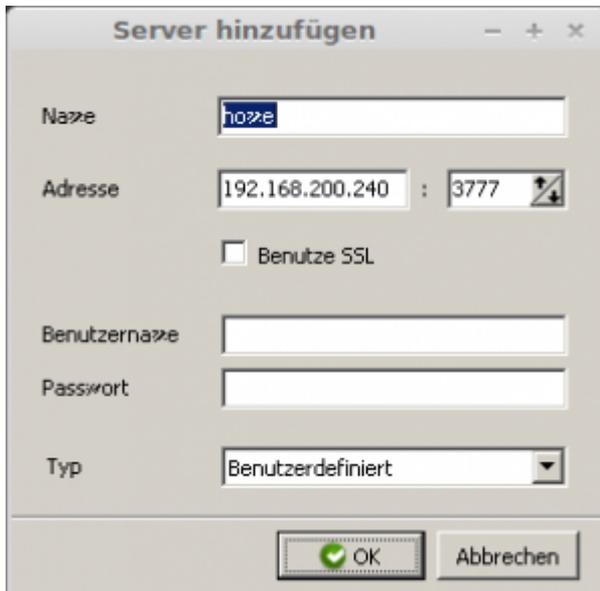
```
sudo service symcon [start / stop / restart]
```

manuell gesteuert werden.

Client Console

Nachdem wir den symcon Dienst nun Installiert und gestartet haben können wir uns mit der Remote Verwaltungskonsole zum Server verbinden. Die Verwaltungskonsole wird am Client gestartet und diese findet ihr unter folgendem Link: [ips_console.exe](https://www.lxccu.com/doku.php?id=manuals:ip-symcon_migration#ips_console_exe)

Nach dem Starten dieser müsst ihr noch die Benutzerdefinierte Verbindung wie folgt einrichten:



Server hinzufügen

Name: hoze

Adresse: 192.168.200.240 : 3777

Benutze SSL

Benutzername:

Passwort:

Typ: Benutzerdefiniert

OK Abbrechen

Solltet ihr auch mit einem Linux System als Client arbeiten und dort die Remote Verwaltungskonsole ausführen so führt folgende Schritte auf eurem Linux Client aus:

```
sudo apt-get update && sudo apt-get install wine winetricks
```

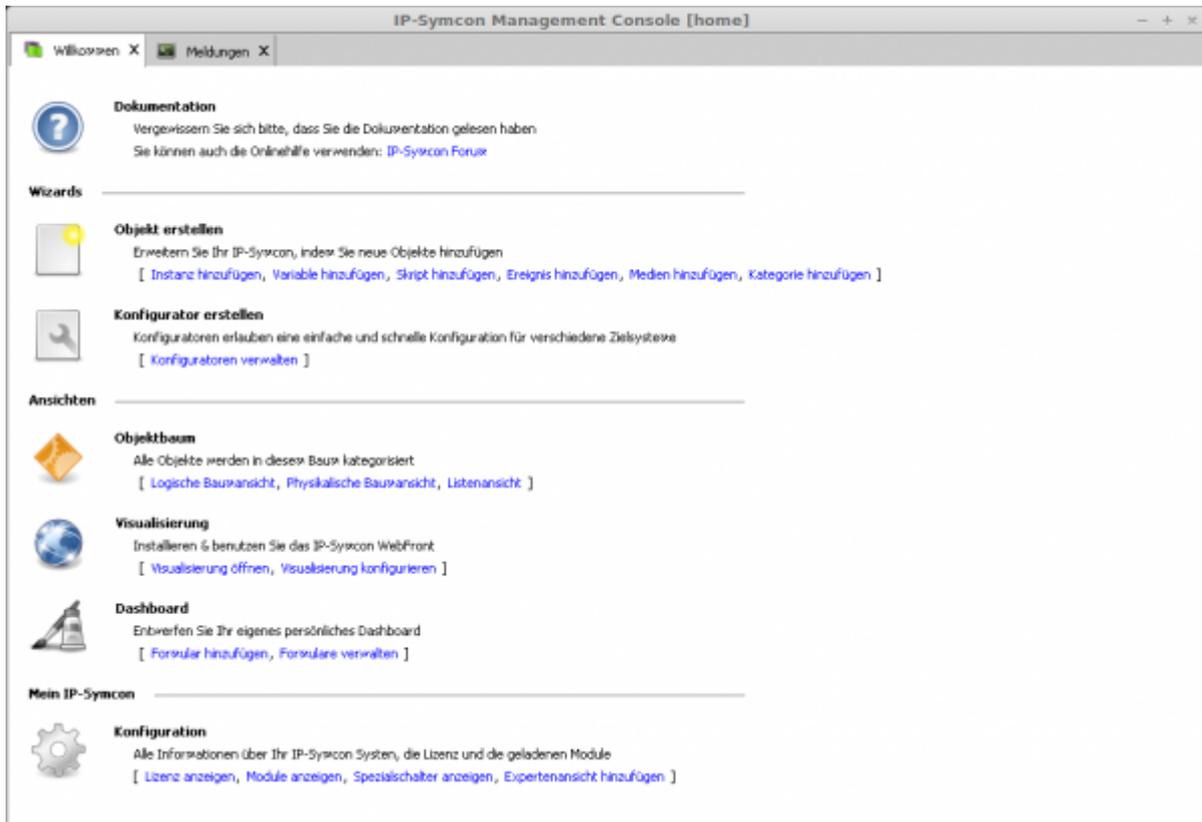
dann die fehlenden Pakete installieren:

```
sudo winetricks msxml3  
sudo winetricks gecko  
sudo winetricks corefonts  
sudo winetricks volnum  
sudo winetricks vcrun2008
```

Nun kann einfach mit:

```
sudo wine /pfad/zur/ips_console.exe /<ip von symcon server>:3777
```

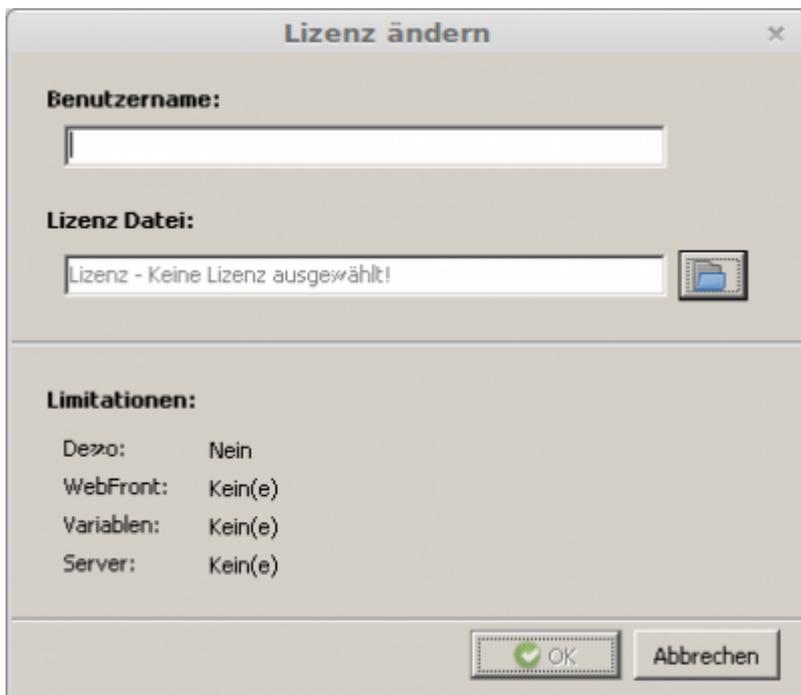
Die Konsole gestartet werden.



Konfiguration

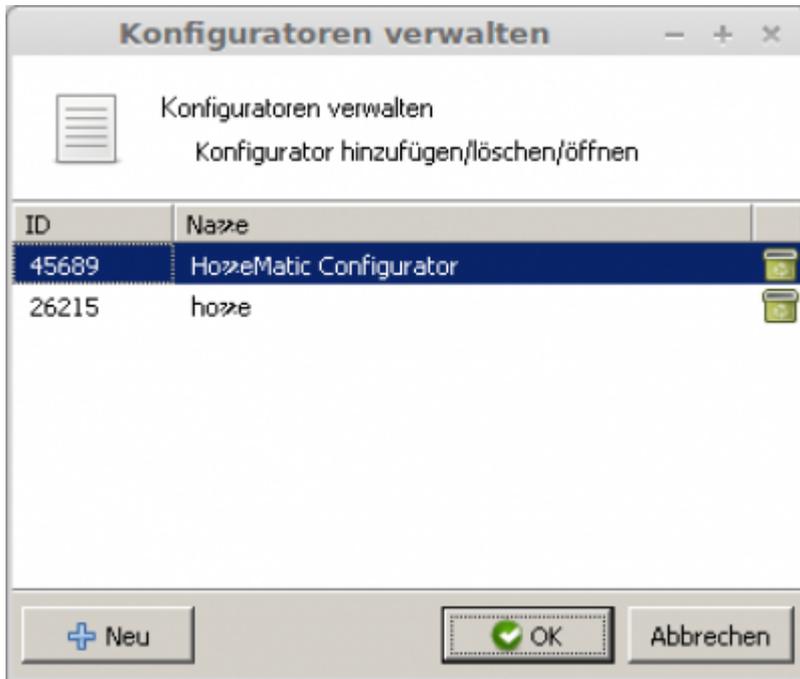
Installieren der Lizenz

In der Konsole einfach mit dem Assistenten den Benutzernamen / Email Adresse eingeben und die Lizenz Datei auswählen:

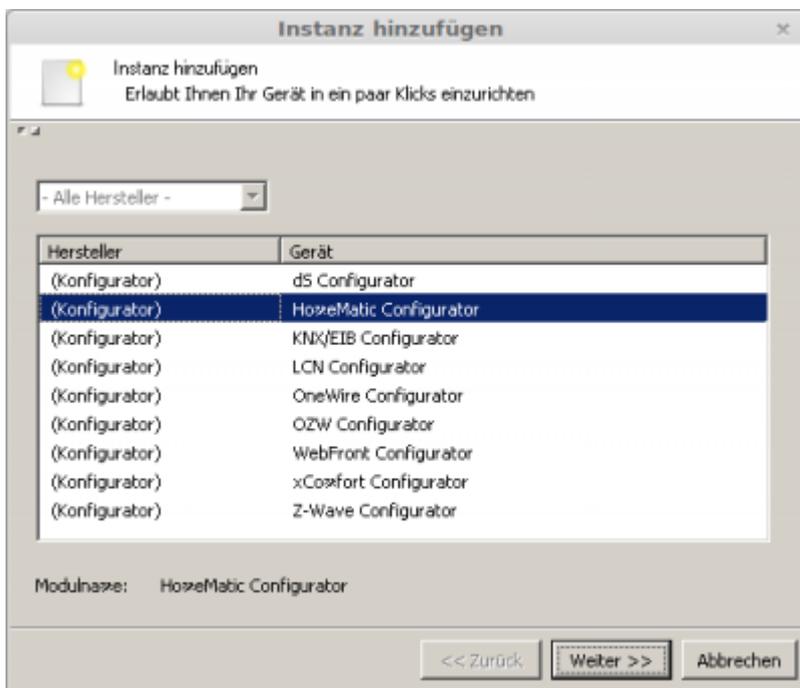


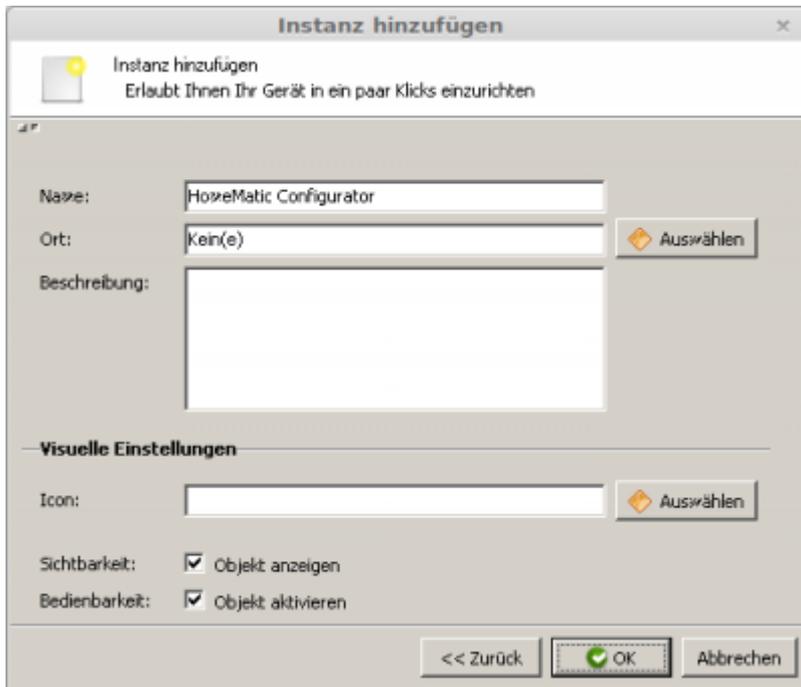
Hinzufügen der CCU

In Der Konsole auf „Konfiguratoren verwalten“ Klicken

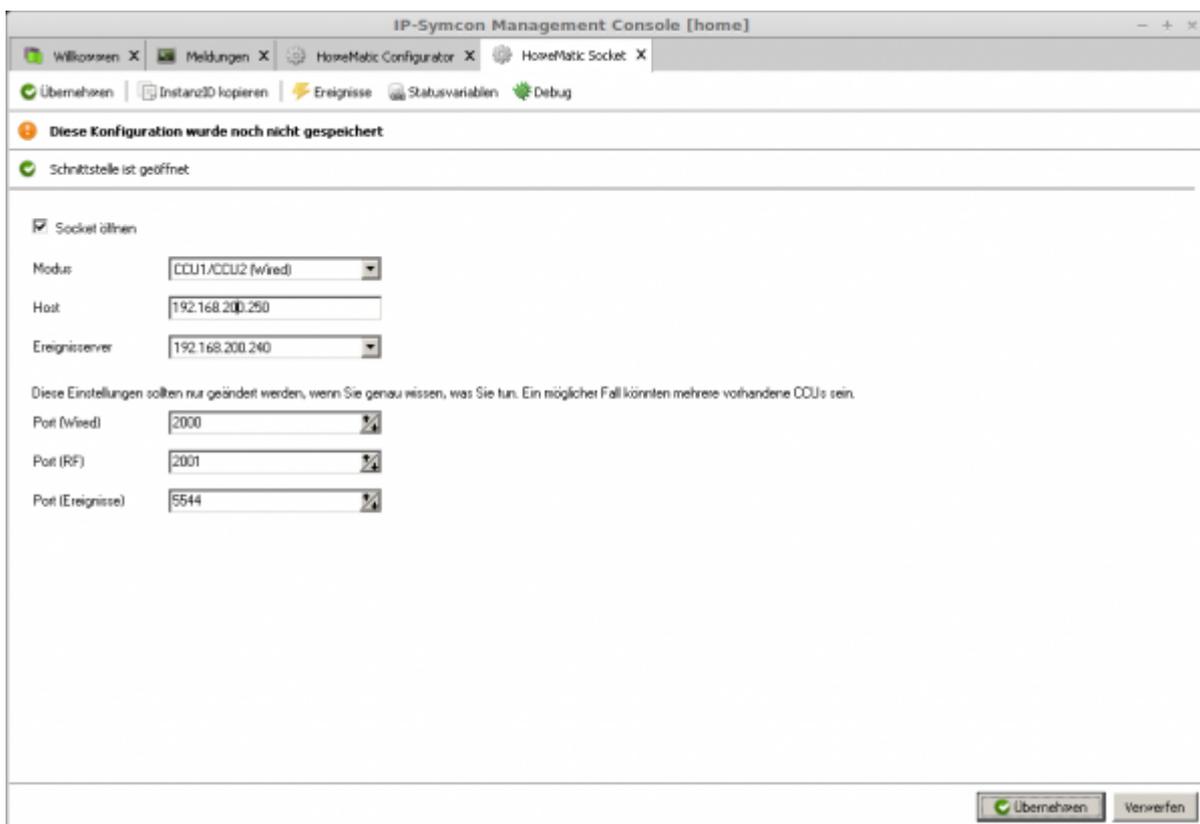


hier auf [NEU] Klicken und dann Homematic Auswählen:





Die IP Adresse der CCU sowie des IP Symcon Server's (Ereignisserver) Eingeben / überprüfen:



Nach Klicken auf Übernehmen sollten nun die auf der CCU verfügbaren Geräte angezeigt werden.

Geräte aus der CCU in IP Symcon hinzufügen

Standardmäßig werden nicht alle Geräte der CCU in IP Symcon angelegt, dies muss manuell für jedes Gerät erfolgen. Auch übernimmt IP Symcon nur die Hometown Adresse (Seriennummer) und nicht den Namen des Gerätes, die Benennung in IP Symcon muss nach dem Erstellen manuell erfolgen.

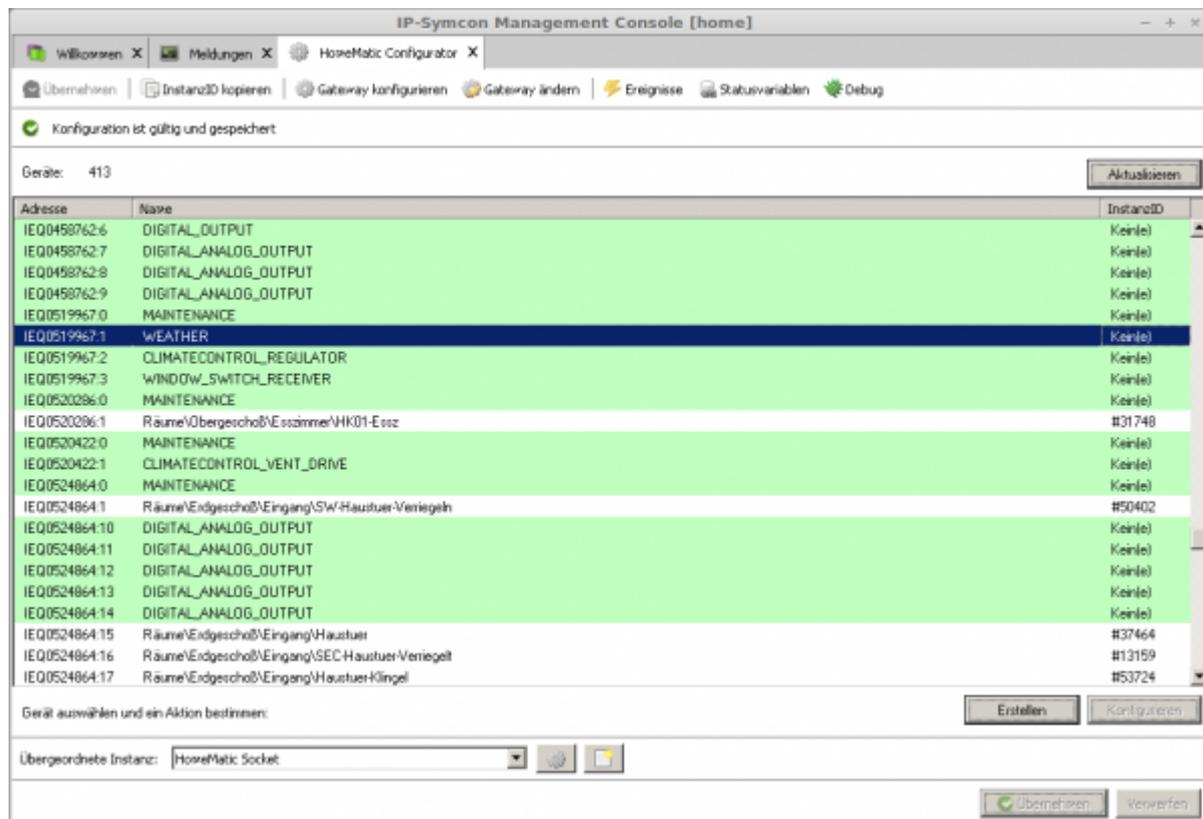
Da IP Symcon die Lizenz über die Anzahl der Variablen Berechnet macht es durchaus sinn nicht alle Geräte und vor allem nicht alle Kanäle der Geräte in IP Symcon anzulegen! Z.B. ein Raumthermostat hat vier Kanäle, der erste mit den aktuellen Temperatur / Luftfeuchte Werten und der Zweite mit der Klima Regelung und der dritte mit Service (Batterie) und dem Sollwert machen sinn aber der vierte mit dem Fensterkontakt für mich nicht womit Variablen „gespart“ werden.



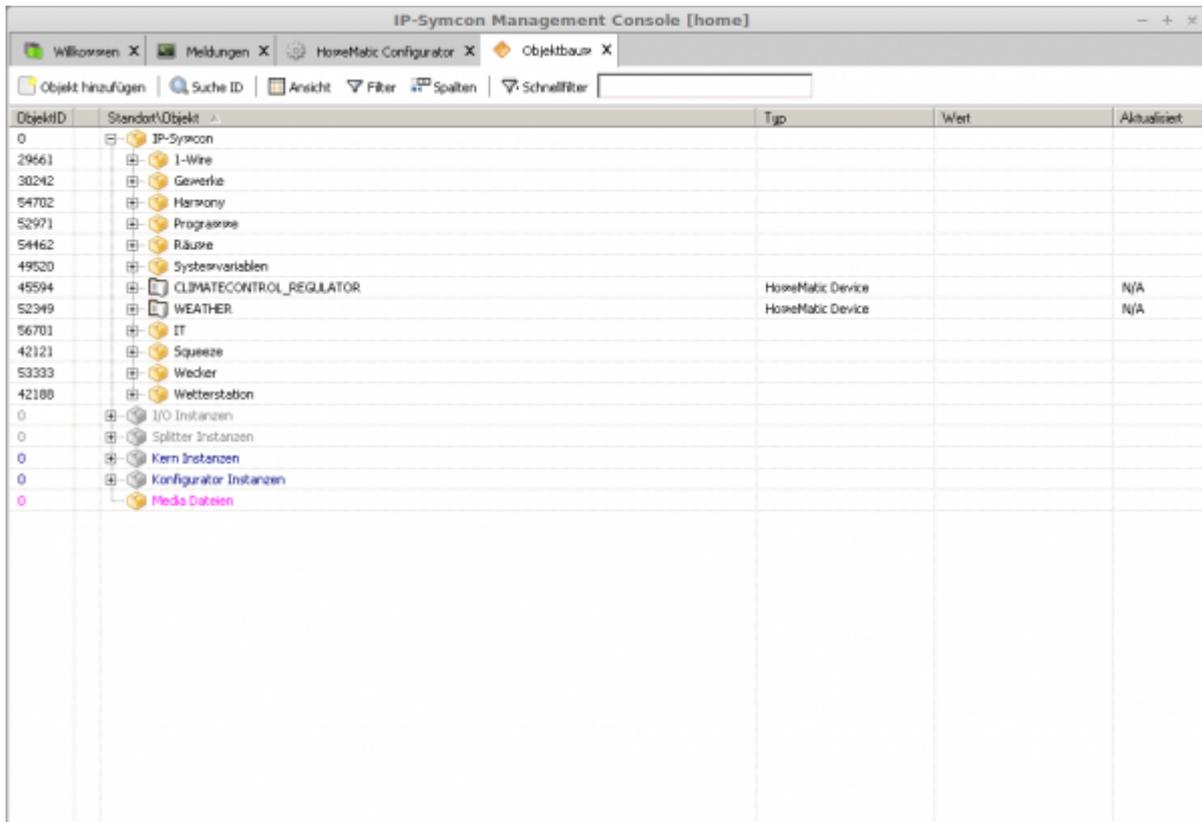
Bei mir benötigt ein Raum mit Heizungsregelung und Rollladen sowie Fensterkontakt 27 Variablen, sollte noch Licht dazu kommen sind es natürlich je nach verwendeten Aktor und in IP Symcon erstellten Kanal mehr. Aber ich habe bei allen Batterie Betrieben Komponenten auch den MAINTENANCE Kanal in IP Symcon erstellt (in dem oben Beschriebenen Raum sind es drei die jeweils 6 Variablen benötigen) weil ich den Batteriestatus damit Überwachen / Dokumentieren kann, außerdem sind hier auch die RSSI (Empfangswerte) und diese können so auch einfach bei Problemen Aufgezeichnet werden.

Je nach Gerät werden bei der Erstellung aller Kanäle bis zu dreizehn Variablen (Schaltaktor mit Leistungsmessung) benötigt!

Am einfachsten sie öffnen das CCU Webinterface und dort die Einstellungen / Geräte Seite, hier sehen sie in der Spalte „Seriennummer“ des gewünschten Gerätes und können dieses in der IP Symcon Verwaltungskonsole suchen und dann dort auf Erstellen Klicken.



Die soeben erstellten Geräte werden dann in der IP Symcon Konsole unter Objektbaum / Logische Baumansicht angezeigt:



ObjektID	Standort/Objekt	Typ	Wert	Aktualisiert
0	IP-Symcon			
29661	1-Wire			
30242	Gewerke			
54702	Harmony			
52971	Programme			
54462	Räume			
49520	Systemvariablen			
45594	CLIMATECONTROL_REGULATOR	HoseMatic Device		N/A
52349	WEATHER	HoseMatic Device		N/A
56701	IT			
42121	Squeeze			
53333	Wecker			
42188	Weckerstation			
0	I/O Instanzen			
0	Splitter Instanzen			
0	Kern Instanzen			
0	Konfigurator Instanzen			
0	Media Dateien			

Dort können wir diese nun einfach Umbenennen sowie in die Kategorien verschieben.



Wie überall Empfohlen habe ich auch eine Kategorie Struktur nach Räume / Geschoß / Raum erstellt. Es dürfen auch die selben Namen in verschiedenen Kategorien verwendet werden, das macht auch sinn weil jeder Raum einen Raumthermostat hat und durch die Struktur seine Zugehörigkeit ersichtlich ist.

Mögliche Konfiguration / Darstellung eines Raumes

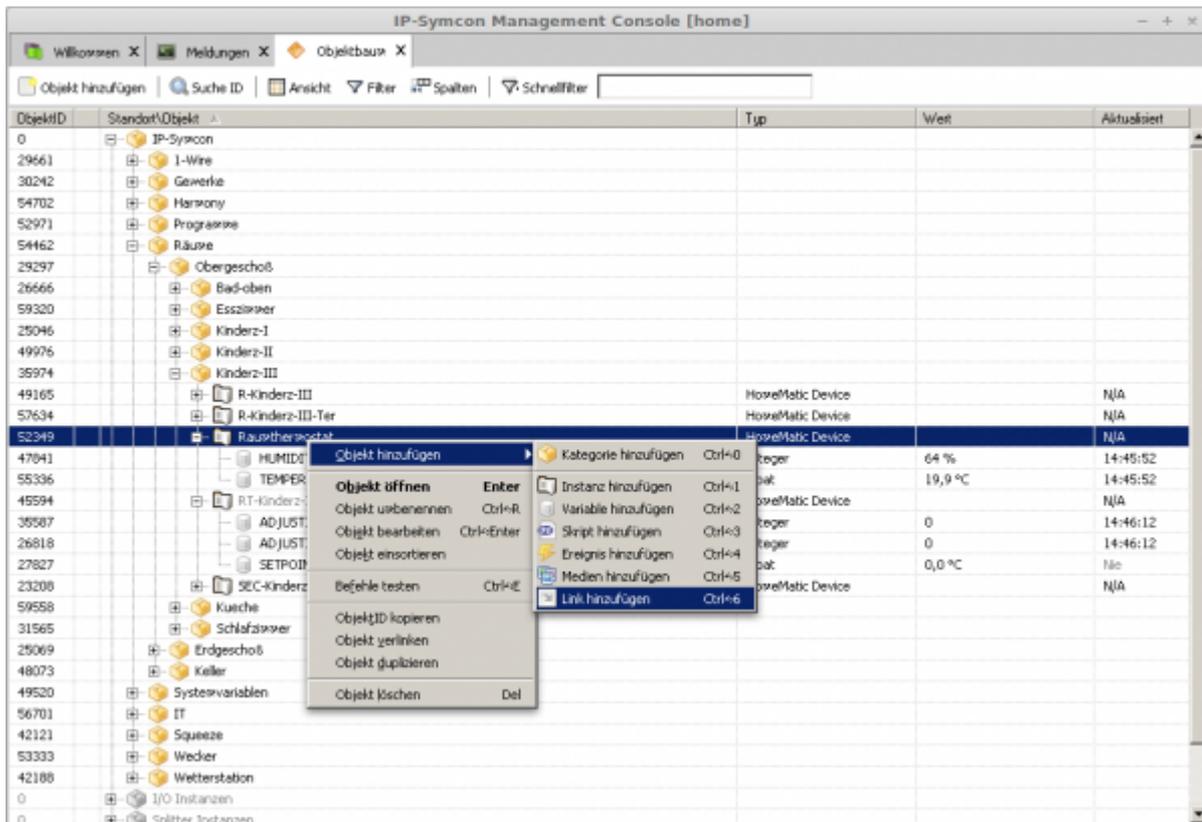
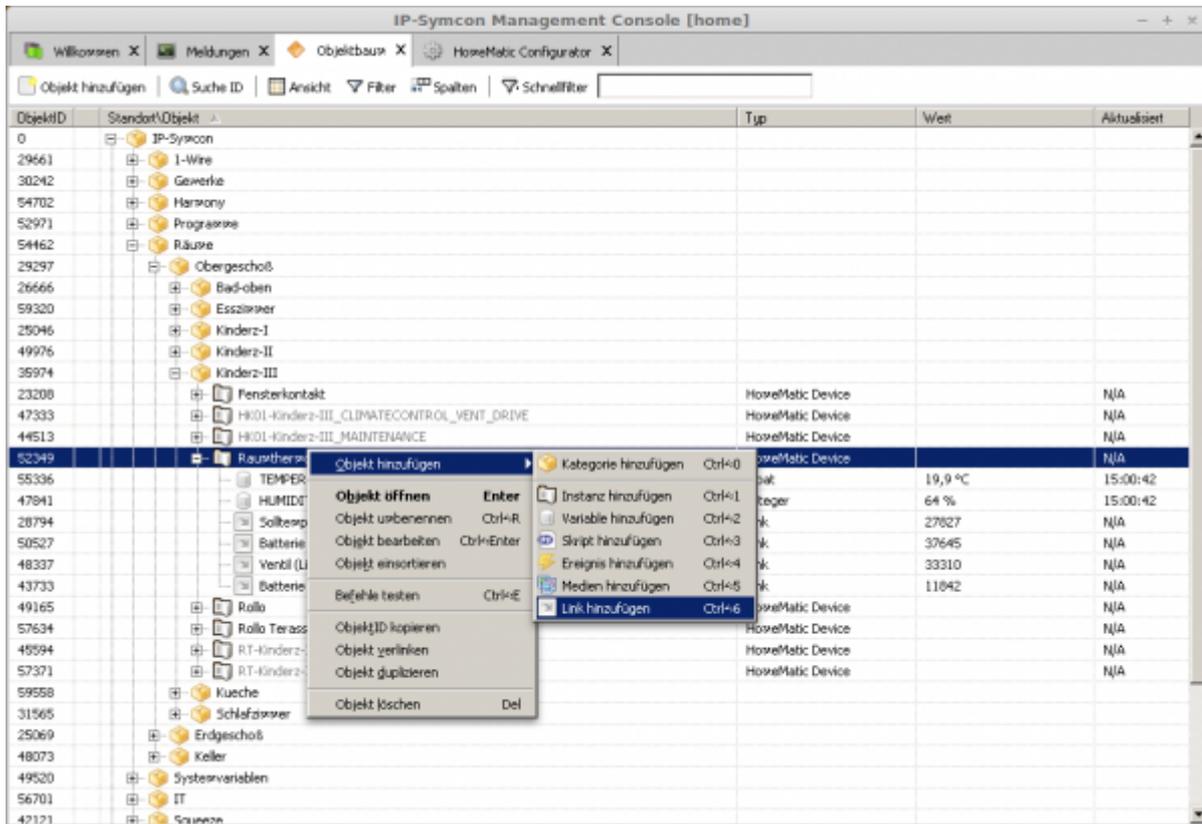
Auch können wir die Geräte eines Raumthermostats zusammenfassen indem wir z.B das CLIMATECONTROL_REGULATOR Ausblenden:

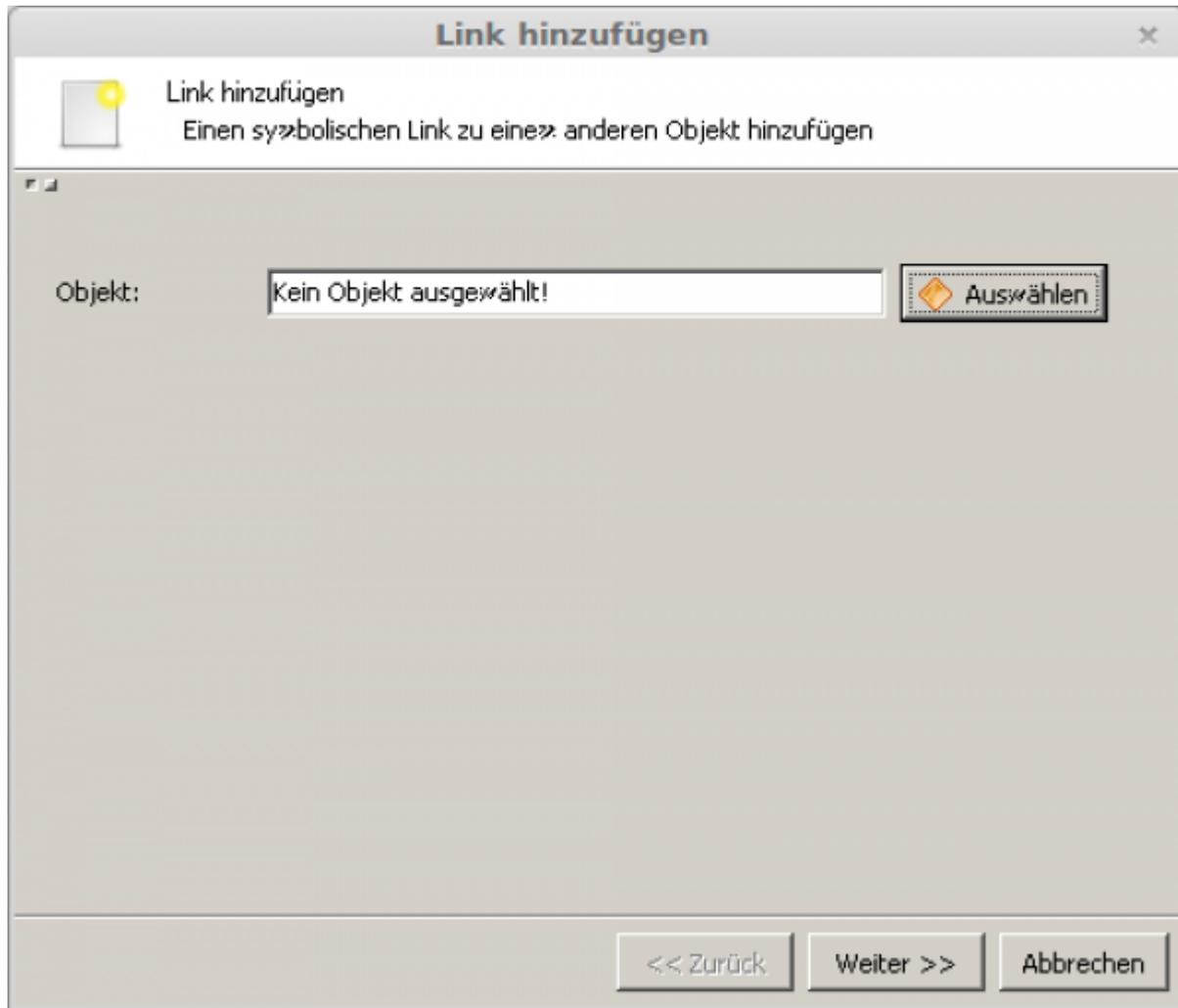
The screenshot shows the IP-Symcon Management Console interface. On the left is a tree view of the object hierarchy. The selected object is 'RT-Kinderz-III_CLIMATECONTROL_REGULATOR'. A context menu is open over this object, listing actions such as 'Objekt hinzufügen', 'Objekt öffnen', 'Objekt umbenennen', 'Objekt bearbeiten', 'Objekt einsortieren', 'Befehle testen', 'ObjektID kopieren', 'Objekt verlinken', 'Objekt duplizieren', and 'Objekt löschen'. On the right is a table with columns: 'ObjektID', 'Standard/Objekt', 'Typ', 'Wert', and 'Aktualisiert'. The table lists various objects and their properties, including 'HUMIDITY' (64 %) and 'TEMPERATURE' (19,8 °C).

The 'Objekt bearbeiten' dialog box is shown. It contains the following fields and options:

- Name:** RT-Kinderz-III_CLIMATECONTROL_REGULATOR
- Ort:** Räume\Obergeschoß\Kinderz-III (with an 'Auswählen' button)
- Beschreibung:** (empty text area)
- Visuelle Einstellungen:**
 - Icon:** (empty field with an 'Auswählen' button)
 - Sichtbarkeit:** Objekt anzeigen
 - Bedienbarkeit:** Objekt aktivieren
- Buttons:** OK and Abbrechen

und nur den Datenpunkt SETPOINT im Raumthermostat anzeigen:





ObjektID	Standort\Objekt	Aktualisiert
29297	Obergeschoß	
26666	Bad-oben	
59320	Essziimmer	
25046	Kinderz-I	
49976	Kinderz-II	
35974	Kinderz-III	
49165	R-Kinderz-III	N/A
57634	R-Kinderz-III-Ter	N/A
52349	Raucherostat	N/A
45594	RT-Kinderz-III_CLIMATECONTROL_RE...	N/A
35587	ADJUSTING_COMMAND	14:48:23
26818	ADJUSTING_DATA	14:48:23
27827	SETPOINT	Nie
23208	SEC-Kinderz-III	N/A
59558	Kueche	
31565	Schlafzimmer	
25069	Erdgeschoß	
48073	Keller	
49520	Systemvariablen	
56701	IT	
42121	Squeeze	
53333	Wecker	
42188	Wetterstation	
0	I/O Instanzen	
0	Splitter Instanzen	
0	Kern Instanzen	

Suche ID | Schnellfilter

Link hinzufügen

Link hinzufügen

Einen symbolischen Link zu einem anderen Objekt hinzufügen

Name:

Ort: Auswählen

Beschreibung:

Visuelle Einstellungen

Icon: Auswählen

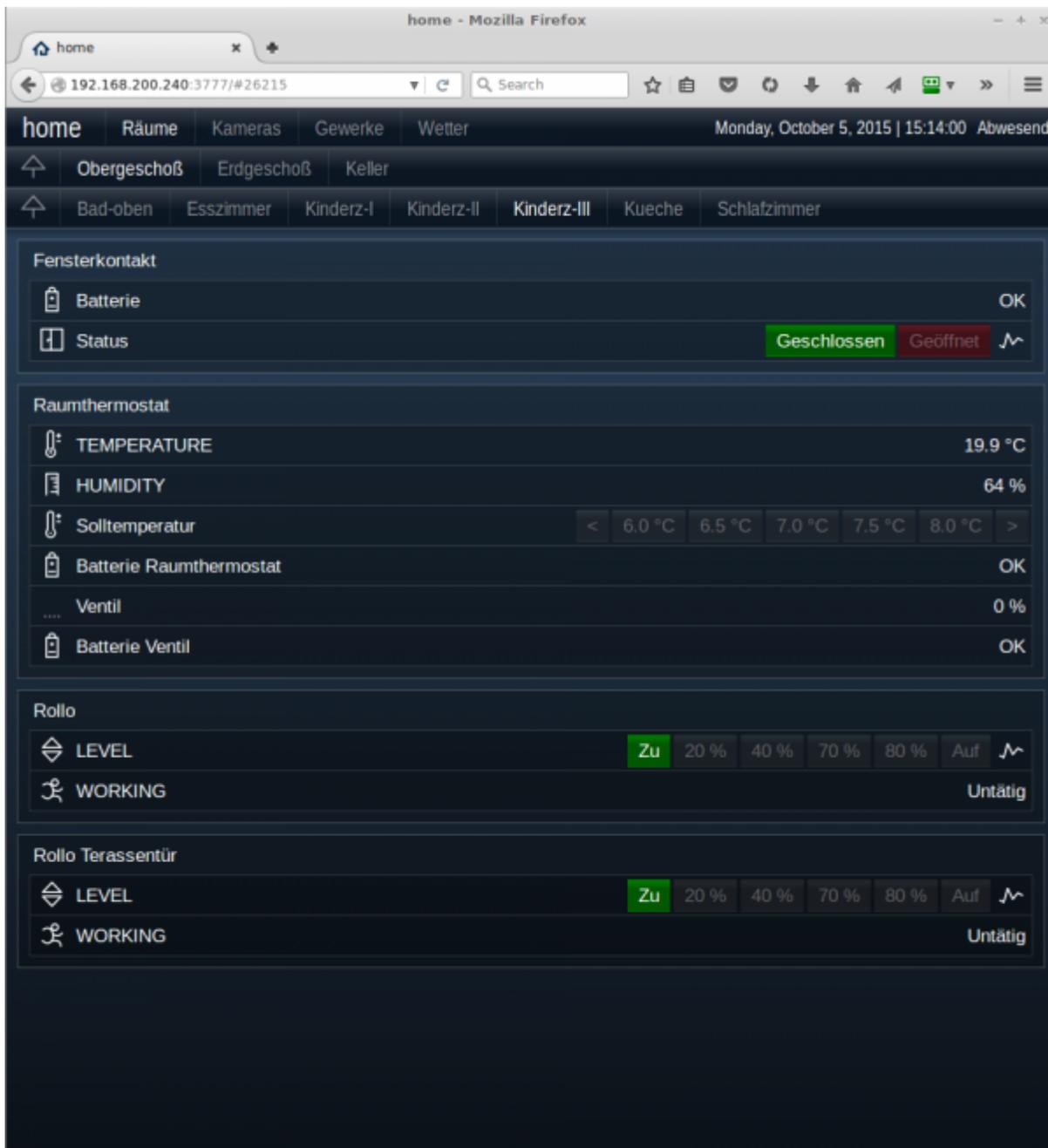
Sichtbarkeit: Objekt anzeigen

Bedienbarkeit: Objekt aktivieren

<< ZurückOKAbbrechen

Hier als Beispiel die Einstellungen der Variablen in der Konsole mit der daraus resultierenden Ansicht des Web Interfaces:

ObjektID	Standort\Objekt	Typ	Wert	Aktualisiert
0	IP-Symcon			
29661	1-Wire			
30242	Gewerke			
54702	Harmony			
52971	Programme			
54462	Räume			
29297	Obergeschoß			
26666	Bad-oben			
59320	Esszimmer			
25046	Kinderz-I			
49976	Kinderz-II			
35974	Kinderz-III			
23208	Fensterkontakt	HomeMatic Device		N/A
57915	Batterie	Boolean	OK	07:01:30
38739	ERROR	Integer	0	07:01:30
43621	INSTALL_TEST	Boolean	False	Ne
17345	Status	Boolean	Geschlossen	07:01:30
47333	HK01-Kinderz-III_CLIMATECONTROL_VENT_DRIVE	HomeMatic Device		N/A
44513	HK01-Kinderz-III_MAINTENANCE	HomeMatic Device		N/A
52349	Raumthermostat	HomeMatic Device		N/A
55336	TEMPERATURE	Float	19,9 °C	15:13:44
47841	HUMIDITY	Integer	64 %	15:13:44
28794	Solltemperatur (Link: Räume\Obergeschoß\Kinderz-III\RT-Kinderz-III_CL...	Link	27827	N/A
50527	Batterie Raumthermostat (Link: Räume\Obergeschoß\Kinderz-III\RT-Kind...	Link	37645	N/A
48337	Ventil (Link: Räume\Obergeschoß\Kinderz-III\HK01-Kinderz-III_CLIMATE...	Link	33310	N/A
43733	Batterie Ventil (Link: Räume\Obergeschoß\Kinderz-III\HK01-Kinderz-III...	Link	11842	N/A
49165	Rollo	HomeMatic Device		N/A
16136	DIRECTION	Integer	0	04:43:03
32015	INHIBIT	Boolean	False	Ne
21309	LEVEL	Float	Zu	04:43:03
29059	WORKING	Boolean	Untäb	04:43:03
57634	Rollo Terrassentür	HomeMatic Device		N/A
40767	DIRECTION	Integer	0	04:40:03
11539	INHIBIT	Boolean	False	Ne
39370	LEVEL	Float	Zu	04:40:03
41658	WORKING	Boolean	Untäb	04:40:03
45594	RT-Kinderz-III_CLIMATECONTROL_REGULATOR	HomeMatic Device		N/A
57371	RT-Kinderz-III_MAINTENANCE	HomeMatic Device		N/A
59558	Kueche			
31565	Schlafzimmer			
25069	Erdschoß			
48073	Keller			
49520	Systemvariablen			
56701	IT			
42121	Squeeze			
53333	Wedkar			
42188	Wetterstation			
0	I/O Instanzen			
0	Splitter Instanzen			
0	Kern Instanzen			
0	Konfigurator Instanzen			
0	Media Dateien			



Systemvariablen Übernehmen

Da die CCU keine Möglichkeit anbietet mit RPC die System variablen auszulesen muss dies Manuell gepollt werden.

Das hört sich schlimmer an als es ist und auch hat keine andere Software eine bessere Lösung an die Systemvariablen zu kommen.

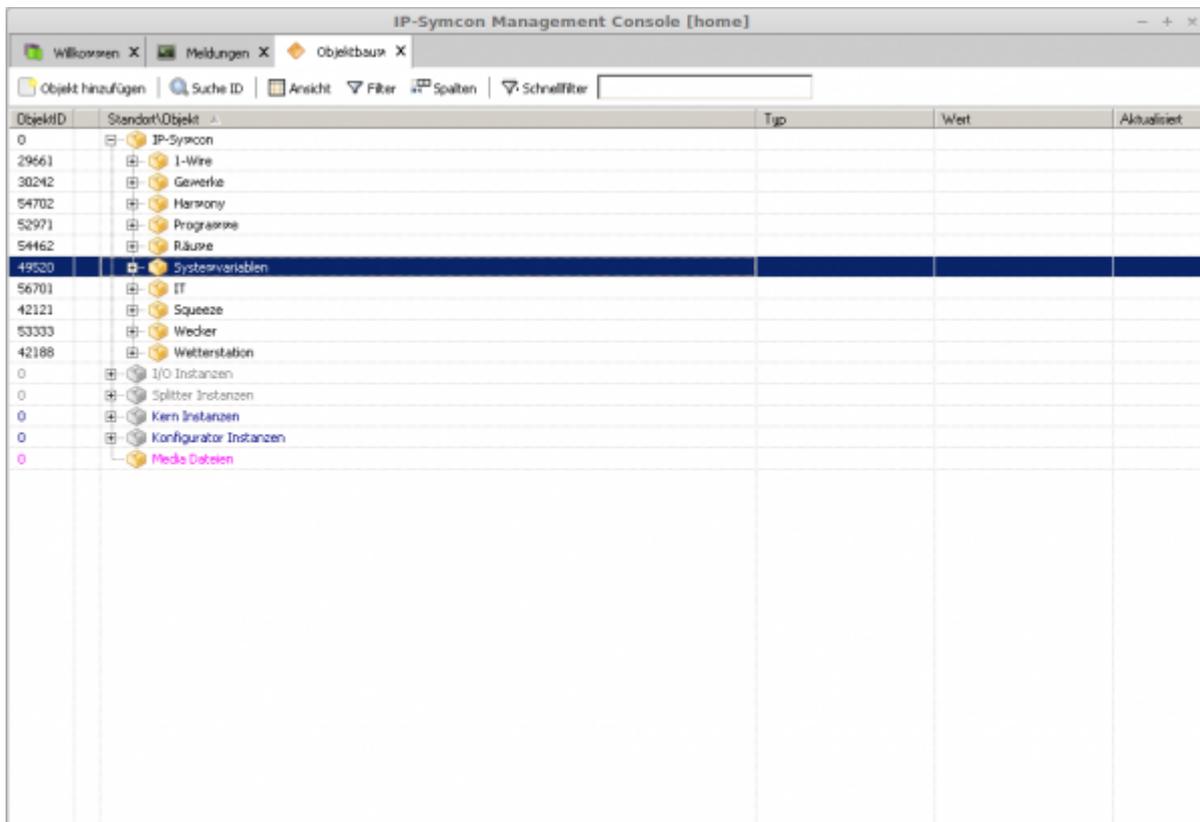


Wenn es das [hier](#) vorgestellte Modul auch für Linux gibt ist dies dem folgenden vorzuziehen! Weil dies ohne XML AddOn sowohl die System variablen aus der CCU lesen und auch schreiben kann sowie auch alle Programme der CCU in IPS anzeigt und Ausführbar macht.

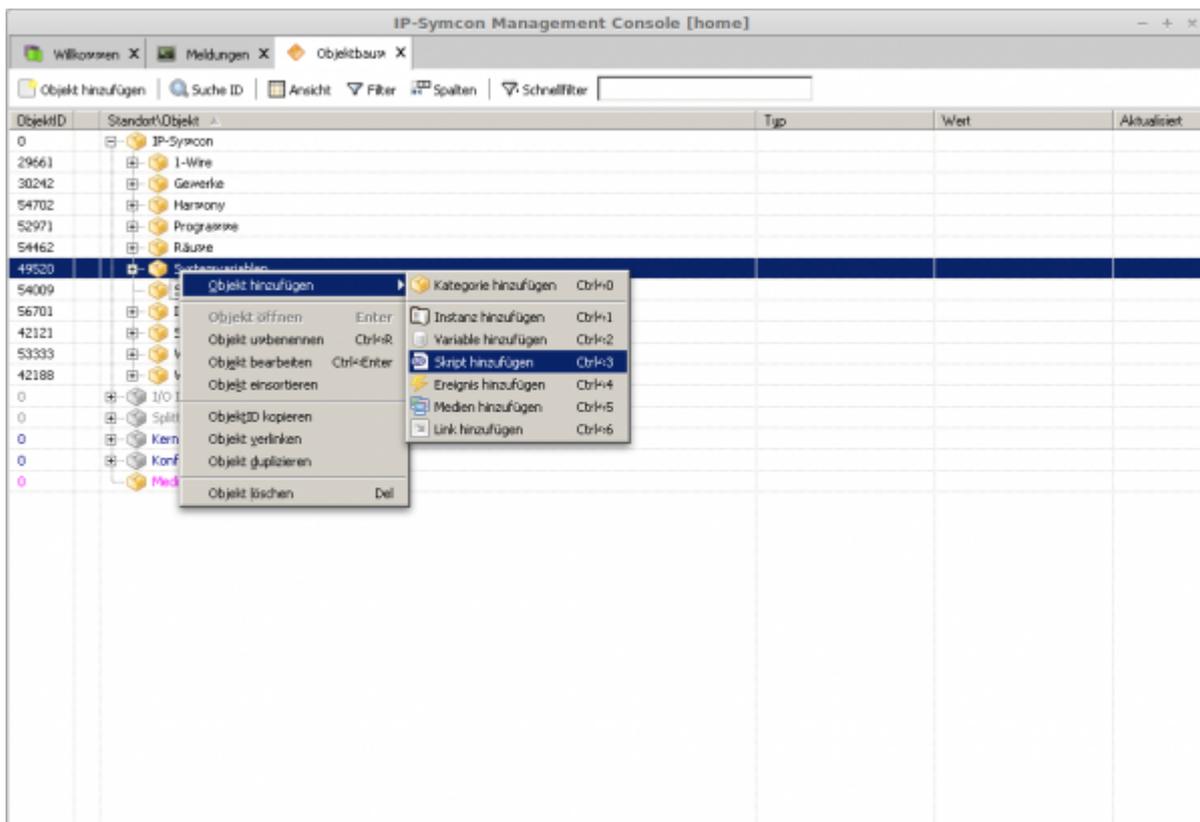
Ich habe den weg über das bei mir in der CCU bereits Installiert AddOn XML-API gewählt, dazu muss in der CCU eben dieses Addon Installiert werden. Wie das geht findet ihr [hier](#)

Die Informationen dazu habe ich von [hier](#).

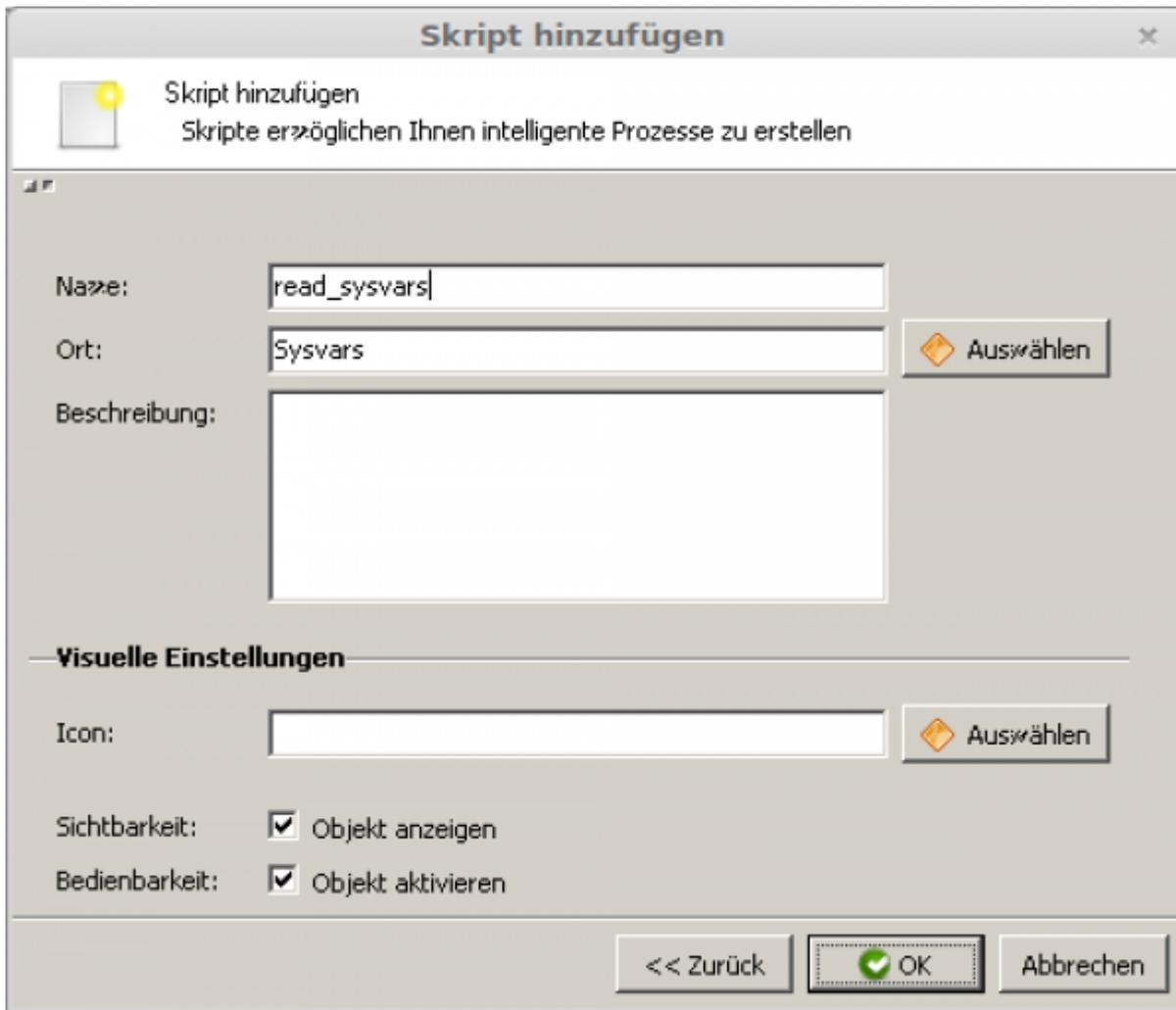
Wir legen eine Kategorie „Systemvariablen“ im Objektbaum an:



Hier ein neues Script:



dem wir den namen „read_sysvars“ geben:



dann den Inhalt aus dem Link oben

https://www.symcon.de/wiki/Systemvariablen#Mit_installierten_XML-API_Patch einfügen und die IP Adresse der CCU so wie im folgenden Bild dargestellt eingeben:

```
1 <?
2 // IP-Adresse der CCU kann über die folgende drei Wege festgelegt bzw. vorgegeben werden
3 $ccu_ip = '192.168.200.250';
4 $par_id = IPS_GetParent($_IPS['SELF']);
5
6 // Timer oder Eventgesteuert ?
7 // 1 - Timer
8 // $event = false;
9 // 2 - Event;
10 $event = 10202 /* ID der virtuellen FB von der CCU welche das Script triggern soll */;
11
12 // Uhrzeit von IPS und CCU müssen gleich sein, wenn das nicht sichergestellt ist, folgende Zeile aktivieren.
13 // $getall = true;
14
15 //-----ENDE KONFIG -- ab hier nichts mehr ändern-----//
16 if ($_IPS['SENDER'] == 'Execute')
17 {
18     if (!isset($event)) die ("Kein Event und kein Timer gesetzt!");
19     print "Install...\n";
20
21     if ($event == false)
22     {
23         if (IPS_GetScriptTimer($_IPS['SELF']) == 0) IPS_SetScriptTimer($_IPS['SELF'],300);
24     } else {
25         if (count(IPS_GetScriptEventList($_IPS['SELF'])) == 0)
26         {
27             $eid = IPS_CreateEvent(0);
28             IPS_SetEventTrigger($eid, 0, $event);
29             IPS_SetParent($eid, $_IPS['SELF']);
30             IPS_SetEventActive($eid, true);
31         }
32     }
33 }
34 $vars_old = IPS_GetChildrenIDs($par_id);
35 $vars_new = Array();
36 $olderr=error_reporting();
37 error_reporting(0);
38 $xml = simplexml_load_file('http://'.$ccu_ip.'/config/xmlapi/sysvarlist.cgi');
```

Nun können wir das Script manuell Starten über das Kontex Menü:

ObjektID	Standard/Objekt	Typ	Wert	Aktualisiert
46291	SYS_Bew_Hochzeit	Boolean	False	15:32:01
37210	SYS_Bew_Wiese	Boolean	False	15:32:01
20715	SYS_ETA_PUSH_MSG	String	Puffer_oben48.80 Puffe...	15:32:01
24409	SYS_HK1_Pumpe	String	Auto	15:32:01
31283	SYS_Puffer_unten_alt	Integer	43	15:32:01
57591	SYS_Rollo_Auto_Run	Boolean	False	15:32:01
20768	SYS_Sicherheitsüberprüfung	Boolean	False	15:32:01
10651	SYS_TEST_SOLL	Integer	20	15:32:01
45938	Tageszeit	Integer	5	15:32:01
47359	Tageszeit2	String	Nachmittag	15:32:01
42432	TEMP_Heizung_SOLL	Integer	0	15:32:01
26383	TEST_V1	String	2014.12.08 18:58:23	15:32:01
41322	TEST_V2	String	2014.12.08 18:58:23	15:32:01
34989	Urlaub	Boolean	False	15:32:01
26020	watchdog_besp	Integer	5	15:32:01
29930	Wecker	String	07:00:00	15:32:01
14192	Wecker_pre	String	06:45:00	15:32:01
14051	Wecker_Set	Boolean	False	15:32:01
15666	Wecker_status	Integer	2	15:32:01
16905	wetter_out_humidity	Integer	0	15:32:01
19079	wetter_out_besp	Integer	-50	15:32:01
24011	Wiese_Pumpe_dauer	Integer	15	15:32:01
28410	xBMC_SZ	Boolean	True	15:32:01
10236	read_sysvars	Script	10236-ips.php	15:32:01
29753		Ereignis		15:32:00

Context menu for 'read_sysvars':

- Objekt hinzufügen
- Objekt öffnen (Enter)
- Objekt umbenennen (Ctrl+R)
- Objekt bearbeiten (Ctrl+Enter)
- Objekt einsortieren
- Script ausführen (Ctrl+E)
- ObjektID kopieren

Wenn dies erfolgreich gelaufen ist und die System variablen erstellt wurden können wir einfach ein Zeitgesteuertes Ereignis welches das Script alle 30 Sekunden ausführt hinzufügen:

IP-Symcon Management Console [home]

Objekt hinzufügen | Suche ID | Ansicht | Filter | Spalten | Schnellfilter

ObjektID	Standard/Objekt	Typ	Wert	Aktualisiert
46291	SYS_Bew_Hochzeit	Boolean	False	15:32:31
37210	SYS_Bew_Wiese	Boolean	False	15:32:31
20715	SYS_ETZ_PUSH_MBG	String	Puffer_oben[46.80]Puffe...	15:32:31
24409	SYS_HK1_Pumpe	String	Auto	15:32:31
31283	SYS_Puffer_unken_alt	Integer	43	15:32:31
57591	SYS_Rollo_Auto_Run	Boolean	False	15:32:31
20768	SYS_Sicherheitsueberpruefung	Boolean	False	15:32:31
10651	SYS_TEST_SOLL	Integer	20	15:32:31
45938	Tageszeit	Integer	5	15:32:31
47359	Tageszeit2	String	Nachmittag	15:32:31
42432	TEMP_Heizung_SOLL	Integer	0	15:32:31
26383	TEST_V1	String	2014.12.08 18:58:23	15:32:31
41322	TEST_V2	String	2014.12.08 18:58:23	15:32:31
34989	Urlaub	Boolean	False	15:32:31
26020	watchdog_bew	Integer	5	15:32:31
29930	Wecker_...	String	07:00:00	15:32:31
14192	We...re	String	06:45:00	15:32:31
14051	We...t	Boolean	False	15:32:31
15666	We...Status	Integer	2	15:32:31
16905	wetter_out_humidity	Integer	0	15:32:31
19079	wetter_out_temp	Integer	-50	15:32:31
24011	Wiese_Pumpe_dauer	Integer	15	15:32:31
28418	>BMC_SZ	Boolean	True	15:32:31
10236	event_scripts	Skript	10236-ips.php	15:32:31
29753		Ereignis		15:32:30

Objekt hinzufügen | Kategorie hinzufügen | Instanz hinzufügen | Variable hinzufügen | Skript hinzufügen | Ereignis hinzufügen | Medien hinzufügen | Link hinzufügen

Ereignis hinzufügen

Ereignisse können Skripte bei Änderungen oder auf zyklischer Basis starten

- Ausgelöstes Ereignis**
Skript wird gestartet, wenn ein bestimmtes Ereignis auftritt
- Zyklisches Ereignis**
Skript wird zyklisch gestartet oder zu einem fest definierten Zeitpunkt
- Wochenplan Ereignis**
Das Wochenplan Ereignis erlaubt das Schalten zu unterschiedlichen Zeitpunkten innerhalb der Woche

<< Zurück | Weiter >> | Abbrechen

Ereignis hinzufügen

Ereignis hinzufügen

Ereignisse können Skripte bei Änderungen oder auf zyklischer Basis starten

Datuzust

Kein(e)
 Einmalig
 Täglich
 Wöchentlich
 Monatlich
 Jährlich

Datuzust nicht prüfen

Zeitmuster

Einmalig
 Sekündlich
 Minütlich
 Stündlich

Alle Sekunden Ohne Begrenzung

Seit Bis

<< Zurück Weiter >> Abbrechen

Programme

Beispiel eines Programms das alle „Lichter“ schalten kann. Im Gegensatz zu Homematic wo per Skript alle Geräte eines Gewerks durchlaufen werden können kann in IP Symcon per Script alle Variablen welche ein Profil zugeordnet haben durchgehen. Dies ist unabhängig welches System dahinter steht, also egal ob Homematic, FS20, KNX...

Den Code habe ich [hier](#) gefunden.

```
<?
/* -----
Alle Lichter aus / anschalten

Dieses Script ermittelt alle Lichter anhand eines Profils und schaltet
diese ein bzw. aus.

Wird das Skript normal aufgerufen, schaltet es aus. Über runScriptEX kann
aber auch der Wert mitgegeben werden:
IPS_RunScriptEx(<SCRIPTID>, Array("VALUE"=>true, "FORCE"=>true);

$c_profiles: string, Variablenprofil das geprüft werden soll
```

```

$c_force: true/false, Wenn TRUE wird auf jeden Fall geschalten, egal
           ob der aktuelle Wert in IPS bereits den Zielwert hat.
           Bei Systemen ohne Rückkanal notwendig.
$c_delay: integer, Millisekunden Pause bis zum nächsten Schaltvorgang

-----*/

$c_profiles[] = "Licht";
$c_profiles[] = "Lichter";

$c_force = false; // S.o.
$c_delay = 250;

/* End of configuration -----*/
DEFINE("ZWAVE", "{101352E1-88C7-4F16-998B-E20D50779AF6}");
DEFINE("FS20", "{48FCFDC1-11A5-4309-BB0B-A0DB8042A969}");
DEFINE("HOME_MATIC", "{EE4A81C6-5C90-4DB7-AD2F-F6BBD521412E}");
DEFINE("XCOMFORT", "{1B7B5B7D-CAA9-4AB5-B9D8-EC805EC955AD}");
DEFINE("LCN", "{2D871359-14D8-493F-9B01-26432E3A710F}");

// status Lesen -
$stat = @$_IPS['VALUE'];
settype($stat, "boolean");

$force = (isset($_IPS['FORCE']))? $_IPS['FORCE'] : $c_force;
settype($force, "boolean");

// Alle Lichter ausschalten (mit Profil Licht!)
$a_var = IPS_GetVariableList();
foreach($a_var as $varid){
    $var_info = IPS_GetVariable($varid);
    $var_prof = (!empty($var_info["VariableCustomProfile"]))?
                $var_info["VariableCustomProfile"] :
                $var_info["VariableProfile"];
    $var_data = getValue($varid);

    if(in_array($var_prof,$c_profiles) && (($var_data != $stat) || $force)
){
        // Übergeordnete Instanz ermitteln.
        $instid = IPS_GetParent($varid);
        $inst_info = IPS_GetInstance($instid);
        switch($inst_info["ModuleInfo"]["ModuleID"]){
            case ZWAVE:
                ZW_SwitchMode($instid, $stat);
                break;
            case FS20:
                FS20_SwitchMode($instid, $stat);
                break;

```

```
    case LCN:
        LCN_SwitchMode($instid, $stat);
        break;
    case XCOMFORT:
        MXC_SwitchMode($instid, $stat);
        break;
    case HOME_MATIC:
        HM_WriteValueBoolean($instid, "STATE", $stat);
        break;
} // Switch
IPS_Sleep($c_delay);
} // if
}
?>
```