



Homematic Scriptprogrammierung

Teil 14 – Projekt Lichtszenen (Fortsetzung)

The screenshot shows the 'Skript bearbeiten' (Edit Script) window in the Homematic Admin interface. The window title is 'Skript bearbeiten' and it contains a table with columns for 'Name', 'Beschreibung', and 'Aktion'. The script is titled 'Anwesenheitssimulation' and includes the following code:

```
foreach(itemID, dom.GetObject("Anwesenheitssimulation").EnumUsedIDs())
{
  var item = dom.GetObject(itemID);
  var device = dom.GetObject(item.Device());
  if (item.IsTypeOf(OT_CHANNEL))
  {
    if (device.HssType().Find("HM-LC-Sw")>=0)
    {
      n=n+1;
    }
    if (device.HssType().Find("HM-LC-Dim")>=0)
    {
      n=n+1;
    }
  }
}
```

Skript bearbeiten

```
foreach(itemID, dom.GetObject("Anwesenheitssimulation").EnumUsedIDs())
{
  var item = dom.GetObject(itemID);
  var device = dom.GetObject(item.Device());
  if (item.IsTypeOf(OT_CHANNEL))
  {
    if (device.HssType().Find("HM-LC-Sw")>=0)
    {
      n=n+1;
    }
    if (device.HssType().Find("HM-LC-Dim")>=0)
    {
      n=n+1;
    }
  }
}
```

\$val\$ = \$this\$ = \$src\$ =

Fehlerprüfung

In diesem letzten Teil der Artikelserie schließen wir das Projekt Lichtszenen ab.



In der letzten Folge der Artikelserie wollen wir nun noch eine Änderung an dem Lichtszenensystem einbauen, die das Anlegen der einzelnen Systemvariablen (Zentralenvariablen) zur Speicherung der Werte der Aktoren entfallen lässt. Es wird lediglich noch eine einzige Zentralenvariable benötigt, die die Werte für die Szenen der verschiedenen Taster aufnehmen kann.

Wir nennen dieses Variable VSzeneVal (Typ Zeichenkette).

In diese Variable sollen nun die Werte aller verwendeten Aktoren in Verbindung mit den entsprechenden Szeneschaltern gespeichert werden.

Wir erinnern uns: Der Speichervorgang (Szene <aktuelle Werte> abspeichern) und der Abrufvorgang (Szene einschalten) wurden über 2 Scripte, die von 2 Zentralenprogrammen aufgerufen wurden, gestartet:

Szene speichern:

The screenshot shows the 'Programmerstellung' (Program Creation) screen in the HomeMatic Admin interface. The program is named 'Licht PSzeneSet'. The condition is 'Systemzustand: SzeneSet im Wertebereich größer als 0.00 bei Änderung auslösen'. The script is set to 'Dann...' (Then...) and the script content is 'string itemID; string sElement; string sNew; integer n; stri... sofort'. A red arrow points from the script field to a red box on the right.

Script zum Speichern
der Werte

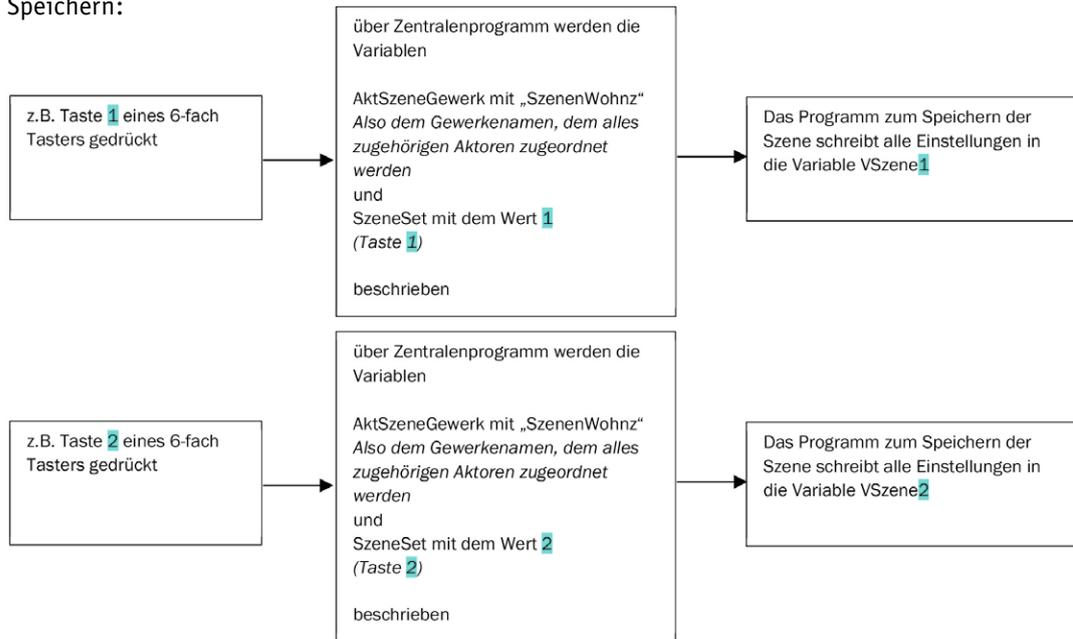
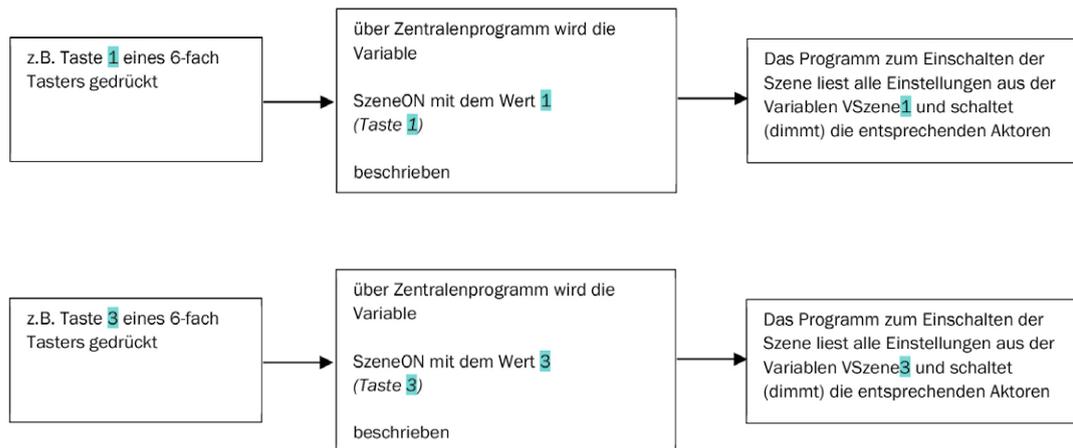
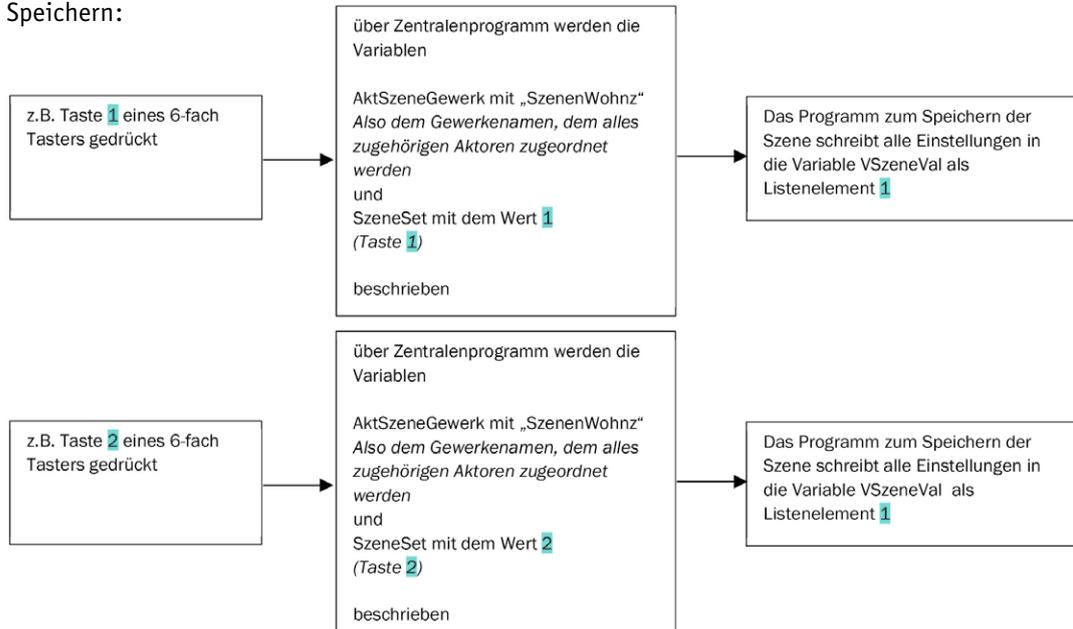
Szene aktivieren/einschalten:

The screenshot shows the 'Programmerstellung' (Program Creation) screen in the HomeMatic Admin interface. The program is named 'Licht PSzeneOn'. The condition is 'Systemzustand: SzeneOn im Wertebereich größer als 0.00 bei Änderung auslösen'. The script is set to 'Dann...' (Then...) and the script content is '!Szene aktivieren !----- dom.GetObject("1... sofort'. A red arrow points from the script field to a red box on the right.

Script zum Einschalten
der Szene

Zum Speichern einer Einstellung wurden in dem Zentralenprogramm für den jeweiligen Schalter 2 Systemvariablen (Zentralenvariablen) gesetzt: die Variable SzeneSet mit der Nummer der Szene und die Variable AktSzeneGewerk mit dem Namen des (Szenen-)Gewerkes.

Zum Einschalten einer Szene wurde lediglich die Variable SzeneOn mit der Nummer der Szene beschrieben.

**Bisher:****Speichern:****Abrufen (Einschalten):****Nach der Änderung:****Speichern:**



Abrufen (Einschalten):



Der Vorteil ist: Es wird eine einzige Systemvariable (Zentralenvariable) zum Speichern der Szenedaten angelegt!

Zur Durchführung der Änderung muss das Datenformat zum Speichern geändert werden. Bisher gab es für jede Szene ein File mit dem Namen VSzene und der Nummer der Szene, also VSzene1, VSzene2 ... u.s.w.

Die Blöcke (pro Szene ein Block), in denen die Daten für die Szene stehen, bleiben grundsätzlich gleich, da aber innerhalb einer Variablen die einzelnen Szenedaten als Liste gespeichert werden, wird das bisher verwendete Komma zum Auflisten der Geräte und ihrer Werte durch das Zeichen # ersetzt und die Aufteilung in mehrere Variablen durch eine Auflistung (,) innerhalb eines Files ersetzt:

Die Daten in VSzeneVal werden also folgendermaßen organisiert:

BLOCK 01 - Entspricht Taster (Szene) mit der Nummer 1													
AKTOR 1		AKTOR 2		AKTOR 3		AKTOR 4		AKTOR 5		AKTOR 6		AKTOR 7	
Item id	Wert/Einstellg	Item id	Wert/Einstellg	Item id	Wert/Einstellg	Item id	Wert/Einstellg	Item id	Wert	Item	Wert	Item id	Wert
14192*0.000000#34492*0.000000#21250*0.000000#13745*0.000000#6814*false#34445*false#34451*false,													
BLOCK 02 - Entspricht Taster (Szene) mit der Nummer 2													
AKTOR 1		AKTOR 2		AKTOR 3		AKTOR 4		AKTOR 5		AKTOR 6		AKTOR 7	
Item id	Wert/Einstellg	Item id	Wert/Einstellg	Item id	Wert/Einstellg	Item id	Wert/Einstellg	Item id	Wert	Item	Wert	Item id	Wert
14192*0.000000#34492*0.000000#21250*1.000000#13745*0.500000#6814*false#34445*true#34451*false,													
BLOCK 03 - Entspricht Taster (Szene) mit der Nummer 3													
AKTOR 2		AKTOR 8		AKTOR 9		AKTOR 10		AKTOR 11		AKTOR 12			
Item id	Wert/Einstellg	Item id	Wert/Einstellg	Item id	Wert/Einstellg	Item id	Wert	Item id	Wert	Item id	Wert		
34492*0.000000,21250*0.000000,13745*0.000000,34246*false,6814*true,34451*false,													

u.s.w.

Die einzelnen Blöcke stellen die Daten für die einzelnen Taster (Szenen) dar. Innerhalb eines Blockes können sich unterschiedlich viele Aktoren befinden, die wiederum durch das Zeichen # getrennt sind. Jeder Aktor wird durch die ID-Nummer und den Wert, der gespeichert wird/ist, beschrieben, wobei hier ID und Wert durch das Zeichen * getrennt sind. Nun müssen noch die beiden Scripte geändert/neu geschrieben werden.

Das erste Script (Abspeichern):

01	!-----
02	!Szene abspeichern
03	!-----
04	string itemID;
05	string sElement;
06	string sNeu;
07	integer n;
08	string sNewBlock;
09	string sOldBlock;
10	integer nAkt;
11	!-----
12	!SzeneschalterNr aus SysVar SzeneSet einlesen und in Zahl iSzSchNr (=Index der Szenedaten in VSzeneVal) wandeln.
13	!-----
14	string sSzSchNr=dom.GetObject("SzeneSet").State().Trunc().ToString();
15	sSzSchNr = sSzSchNr.Substr(0,sSzSchNr.Find("."));



```
16 integer iSzSchNr= sSzSchNr.ToInteger();
17
18 var myAssembly =
  dom.GetObject(dom.GetObject("AktSzeneGewerk").State());
19 string sBlock = "";
20
21
22 !-----
23 !Gewerk Aktoren durchgehen und Werte des Eintrags iSzSchNr in
  sNewBlock merken
24 !-----
25 foreach(itemID, myAssembly.EnumUsedIDs())
26 {
27   var item = dom.GetObject(itemID);
28   if (item.IsTypeOf(OT_CHANNEL))
29   {
30     var device = dom.GetObject(item.Device());
31     if ((device.HssType().Find("HM-LC-Sw") >= 0))
32     {
33       if (sBlock == "")
34       {
35         sBlock = itemID # "*" # item.State();
36       }
37       Else
38       {
39         sBlock = sBlock # "#" # itemID # "*" # item.State();
40       }
41     }
42
43     if ((device.HssType().Find("HM-LC-Dim") >= 0))
44     {
45
46       if (sBlock == "")
47       {
48         sBlock = itemID # "*" # item.State();
49       }
50       else
51       {
52         sBlock = sBlock # "#" # itemID # "*" # item.State();
53       }
54     }
55   }
56 }
57
58
59 !-----
60 !Anzahl der aktuellen Bloecke in VSzeneVal feststellen
61 !-----
62 integer iAnzBlk = 0;
63 foreach (sElement, dom.GetObject("VSzeneVal").State().Split(", "))
64 {
65   iAnzBlk = iAnzBlk + 1 ;
66 }
67
68 !-----
69 !neuen String zum Speichern bilden
70 !-----
71 sNeu="";
72
73 if (iAnzBlk == 0)
74 {
75   if (iSzSchNr > 1)
76   {
77     !-----
78     !bis zur Nummer des Eintrag mit ,,,, beschreiben
79     !-----
80     n=1;
81     while (n<iSzSchNr )
82     {
83       sNeu = sNeu # ", ";
84       n = n +1;
85     }
86     sNeu = sNeu # sBlock;
87   }
88   else
89   {
90     sNeu = sBlock;
91   }
92 }
93
94 Else
95 {
96   sNeu="";
97   nAkt=1;
98   sOldBlock="";
```

Seriennummern und
Zustände merken

VSzeneVal ist noch leer!



99	foreach (sOldBlock, dom.GetObject("VSzeneVal").State().Split(","))	
100	{	
101	if (nAkt<>iSzSchNr)	
102	{	
103	if (sNeu=="")	
104	{	
105	sNeu = sOldBlock;	
106	}	
107	else	
108	{	
109	sNeu = sNeu # "," # sOldBlock	
110	}	
111	}	
112	else	
113	{	
114	if (nAkt==iSzSchNr)	
115	{	
116	if (sNeu=="")	
117	{	
118	sNeu = sBlock;	
119	}	
120	else	
121	{	
122	sNeu = sNeu # "," # sBlock;	
123	}	
124	}	
125	}	
126	nAkt = nAkt + 1;	
127	}	
128	if (iSzSchNr > iAnzBlk)	
129	{	
130	while(nAkt < iSzSchNr)	
131	{	
132	sNeu = sNeu # ",";	
133	nAkt = nAkt + 1;	
134	}	
135	sNeu = sNeu # sBlock;	
136	}	
137	}	
138		
139	!-----	
140	!Neue Werte schreiben	
141	!-----	
142	dom.GetObject("VSzeneVal").State(sNeu);	
143		
144	!-----	
145	!Steuervariablen initialisieren	
146	!-----	
147	dom.GetObject("AktSzeneGewerk").State("");	
148	dom.GetObject("SzeneSet").State(0);	
149		
150		
151		
152		
153		

Es stehen bereits Daten in VSzeneVal !

Die Szenenummer ist höher als die bereits gespeicherten Daten

Das zweite Script (Abrufen/Einschalten):

01	!-----	
02	!Szene aktivieren	
03	!-----	
04		
05		
06	!-----	
07	!SzeneschalterNr aus SysVar SzeneON einlesen und in Zahl	
08	!-----	
09	string sSzSchNr = dom.GetObject("SzeneON").State().Trunc().ToString();	
10	sSzSchNr = sSzSchNr.Substr(0, sSzSchNr.Find("."));	
11	integer iSzSchNr= sSzSchNr.ToInteger();	
12		
13	!-----	
14	!Die entsprechende Aktorenlisite aus dem File holen	
15	!-----	
16	string sListe = dom.GetObject("VSzeneVal").State().StrValueByIndex(", ",	
17		
18		
19	!-----	
20	!Die einzelnen Elemente durchgehen	
21	!-----	
22	string part;	
23		



```

24 foreach(part, sListe.Split("#"))
25 {
26     var itemIDStr = part.Substr(0,part.Find("*"));
27     var item = dom.GetObject(itemIDStr);
28     var valueStr = part.Substr(part.Find("*").ToInteger()+1;
29
30     if ((valueStr != "false") && (valueStr != "true"))
31     {
32         item.State(valueStr);
33     }
34     else
35     {
36         if (valueStr == "false")
37         {
38             item.State(false);
39         }
40         if (valueStr == "true")
41         {
42             item.State(true);
43         }
44     }
45 }
46
47 !-----
48 !aufrufenden Parameter initialisieren
49 !-----
50 dom.GetObject("SzeneOn").State(0);
51

```

Das Zentralenprogramm zum Steuern der Scripte:

The screenshot shows the 'HomeMatic' 'Programmerstellung' (Program Creation) interface. The interface is divided into several sections, each representing a step in the program. The steps are:

- Condition:** 'Wohnzimmer*6 fach Taster*01:T1 bei Tastendruck kur' (Living room*6 switch*01:T1 on button press short).
- Activity:** 'Dann... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).' (Then... Before execution, end all running delays for these activities (e.g., retrigger)).
- Condition:** 'Sonst, wenn...' (Otherwise, when...).
- Activity:** 'Dann... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).' (Then... Before execution, end all running delays for these activities (e.g., retrigger)).
- Condition:** 'Sonst, wenn...' (Otherwise, when...).
- Activity:** 'Dann... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).' (Then... Before execution, end all running delays for these activities (e.g., retrigger)).
- Condition:** 'Sonst, wenn...' (Otherwise, when...).
- Activity:** 'Dann... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).' (Then... Before execution, end all running delays for these activities (e.g., retrigger)).

Red arrows on the right side of the interface point to the following elements:

- Einschalten** (Turn on): Points to the 'Aktivität: Dann...' section where 'SzeneOn' is selected.
- Speichern** (Save): Points to the 'Aktivität: Dann...' section where 'AktSzeneGewerk' and 'SzeneGewerk1' are selected.
- Einschalten** (Turn on): Points to the 'Aktivität: Dann...' section where 'SzeneOn' is selected.
- Speichern** (Save): Points to the 'Aktivität: Dann...' section where 'AktSzeneGewerk' and 'SzeneGewerk1' are selected.

At the bottom of the interface, there are buttons for 'Abbrechen' (Cancel), 'OK', 'Einstellungen als neues Programm speichern' (Save settings as new program), and 'Skript testen' (Test script).

Mit der Möglichkeit, sowohl Szenennummern als auch Szenengewerke beim Speichern einer Szene anzugeben, ist das System nun sehr variabel.



Nicht zu vergessen ist natürlich noch die Systemvariable (Zentralvariable) „Szeneprogrammierung“. Wenn sie gesetzt ist, bewirkt der Druck auf eine Taste eines Szeneschalters das Speichern der aktuellen Zustände der zum aktuellen Szenegewerk gehörigen Aktoren. Ist sie nicht gesetzt, dann schaltet ein Druck auf eine Szenetaste die Szene ein.

Im folgenden Beispiel wird die Programmiervariable durch einen Schlüsselschalter oder durch eine Testvariable (SzeneProgTest) gesetzt und über einen MP3-Funk-Gong wird ein akustisches Signal ausgegeben:

The screenshot shows the HomeMatic Admin interface for configuring a scene program. The breadcrumb trail is: Startseite > Status und Bedienung > Programme und Verknüpfungen > Programme > Programmerstellung. The current page is 'Programme und Verknüpfungen'.

Name	Beschreibung	Bedingung (Wenn...)
Licht SzeneProg		Kanalzustand: Programmierschlüssel:2 bei Tastendruck kurz

Bedingung: Wenn...

- Geräteauswahl: Programmierschlüssel:2 bei Tastendruck kurz
- UND
- ODER
- Systemzustand: SzeneProgTest bei ist wahr bei Änderung auslösen
- UND

Aktivität: Dann... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

- Systemzustand: Szeneprogrammierung sofort ist wahr
- Systemzustand: Szeneprogrammierung verzögert um 5 Sekunden ist falsch
- Geräteauswahl: FunkGong:2 sofort Kanalaktion 1,1,108000,2
- Systemzustand: SzeneProgTest sofort ist falsch

Aktivität: Sonst... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

Mit diesem Artikel endet die Reihe über die Scriptprogrammierung. Themen daraus und weitere Scriptthemen werden Sie ab ELV Journal 2/2019 in einer neuen Artikelreihe finden, in der es um die neue CCU3 in Verbindung mit Mediola-Komponenten geht. **ELV**

Sehr geehrter Leser,

bei diesem Artikel zur Scriptprogrammierung handelt es sich um einen Fachbeitrag eines erfahrenen Homematic Users und Autors. Die ELV/eQ-3 Unternehmensgruppe selbst nutzt die Möglichkeiten dieser Schnittstelle nicht, möchte aber den Anwendern der CCU2 den Zugang zu dieser Schnittstelle nicht verwehren.

Sollten Sie Schwierigkeiten bei der Verwendung dieser zusätzlichen Programmiermöglichkeit der CCU2 haben, so haben Sie bitte Verständnis dafür, dass wir Ihnen hierzu leider keinen Support geben können.

In den entsprechenden Foren und Internetplattformen rund um das Thema „Programmierung Homematic CCU“ finden Sie jedoch sicherlich im Bedarfsfall die notwendigen Anregungen und Hilfestellungen für Ihr Projekt.

Mögliche Quellen im Internet:

<https://www.homematic-inside.de/software/download/item/homematic-skript>

<https://homematic-forum.de/forum/viewtopic.php?f=19&t=18692>