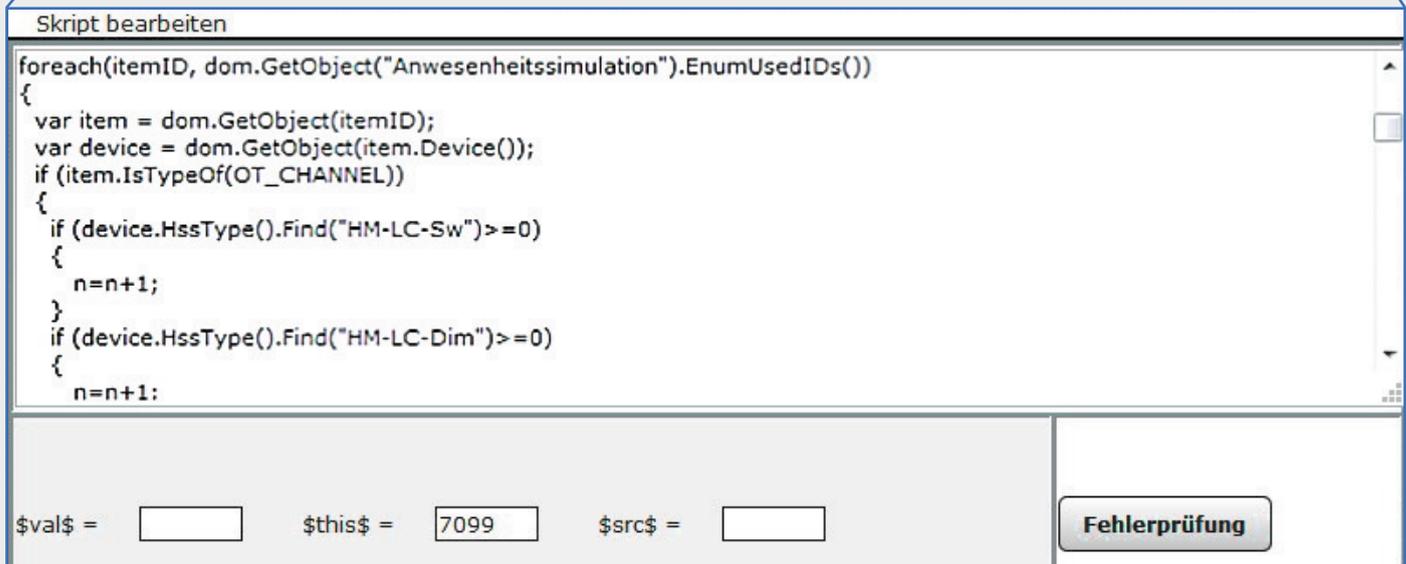
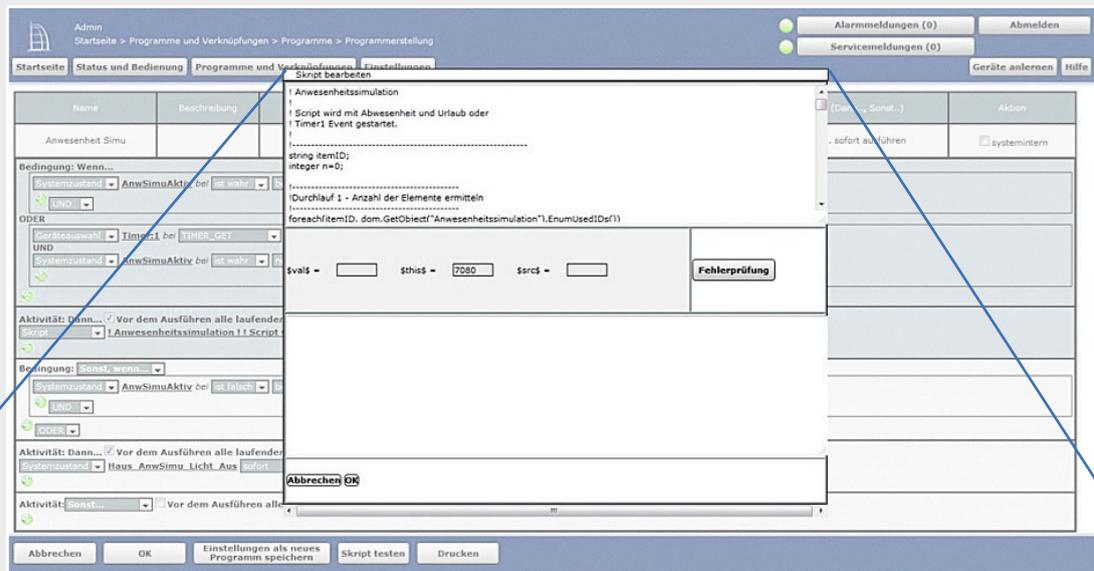




Homematic Scriptprogrammierung

Teil 13 – Projekt Lichtszenen (Fortsetzung)

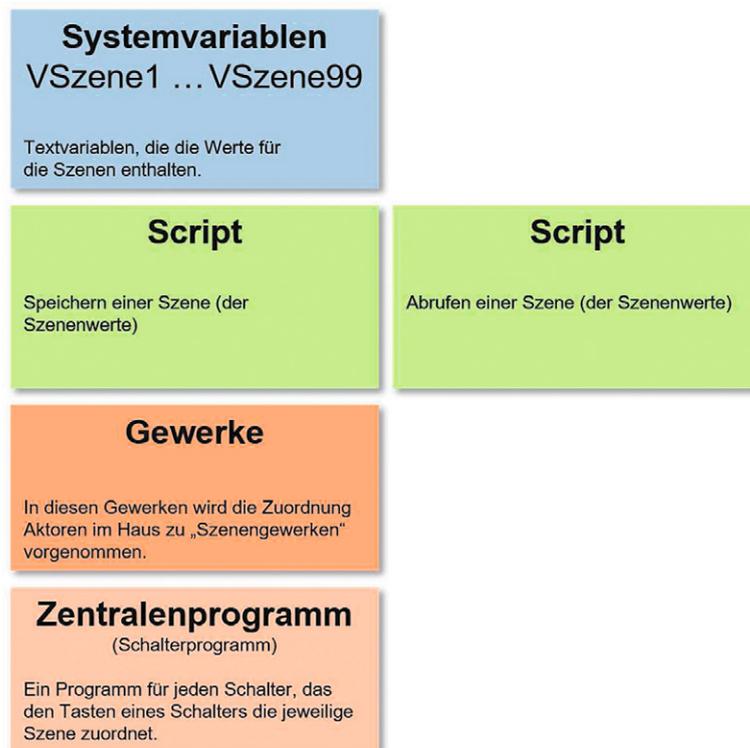


Im dreizehnten Teil der Artikelserie beschäftigen wir uns weiter mit dem Projekt Lichtszenen.



Szenen/Lichtszenen

In Folge 12 der Artikelserie haben wir die Grundlagen für eine komfortable, speicherbare Szenenverwaltung gelegt. Zur Übersicht hier noch mal die Variablen, Programme und Funktionen:



Wenn nun auf mehrere Szenen und Szeneschalter erweitert werden soll, müssen nachfolgende Schritte durchgeführt werden.

Zuerst muss eine Beziehung zwischen den Szeneschaltern und den Gewerken hergestellt werden. Damit bekommt jeder Szeneschalter ein Gewerk zugewiesen, dem wiederum diejenigen Aktoren zugeordnet werden, die von ihm geschaltet werden sollen.

Die Datenspeichernummer (Systemvariable/Zentralenvariable: VSzeneX) wird ja in dem Zentralenprogramm für den Schalter aufgerufen. In dem Script zum Speichern der aktuellen Werte muss aber das zugehörige Gewerk im Scriptprogramm nach den dem Gewerk zugeordneten Aktoren durchgegangen werden (in Teil 12 der Artikelserie wurde hier fest das Gewerk *SzeneGewerk1* verwendet). Um nun unterschiedliche Gewerke (Szeneschalter) möglich zu machen, müssen die Namen der Gewerke in einer Systemvariablen gespeichert werden. Über das Einlesen dieser Systemvariablen kann der Name des Gewerkes dann dem Speicherscript bekannt gegeben werden.

Um uns also die Möglichkeit zu schaffen, den Namen des Gewerkes variabel zu halten, erzeugen wir eine Zentralenvariable mit dem Namen *AktSzeneGewerk*. Und nun müssen wir dafür sorgen, dass in dem Moment, in dem das Script zum Speichern der aktuellen Werte gestartet wird, die Variable (Zentralenvariable) *AktSzeneGewerk* tatsächlich mit dem benötigten Wert beschrieben ist (also vor dem Aufruf beschrieben wurde). Dieses Beschreiben erfolgt im aufrufenden Zentralenprogramm.

Die Variable *AktSzeneGewerk* muss wie gesagt bereits **vor** dem Zeitpunkt, zu dem die Variable *SzeneSet* mit der Nummer der Szene beschrieben wird (um daraufhin den Speichervorgang auszulösen), mit dem Namen des betreffenden Gewerkes beschrieben werden:



Admin
Startseite > Programme und Verknüpfungen > Programme > Programmerstellung

Startseite Status und Bedienung Programme und Verknüpfungen Einstellungen

Systemzustand **SzeneOn** sofort 5.00

Bedingung: **Sonst, wenn...**

Geräteauswahl **Wohnzimmer*6** fach **Taster*01:T1** bei **Tastendruck kurz**

UND

Systemzustand **Szeneprogrammierung** bei **ist wahr** nur prüfen

ODER

Geräteauswahl **Wohnzimmer*6** fach **Taster*01:T1** bei **Tastendruck lang**

UND

Systemzustand **Szeneprogrammierung** bei **ist wahr** nur prüfen

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

Systemzustand **AktSzeneGewerk** sofort **SzeneGewerk1**

Systemzustand **SzeneSet** sofort 1.00

Bedingung: **Sonst, wenn...**

Geräteauswahl **Wohnzimmer*6** fach **Taster*01:T2** bei **Tastendruck kurz**

UND

Systemzustand **Szeneprogrammierung** bei **ist wahr** nur prüfen

ODER

Geräteauswahl **Wohnzimmer*6** fach **Taster*01:T2** bei **Tastendruck lang**

UND

Systemzustand **Szeneprogrammierung** bei **ist wahr** nur prüfen

Abbrechen OK Einstellungen als neues Programm speichern Skript testen Drucken

Sowohl das Initialisieren von *AktSzeneGewerk* als auch das von *SzeneSet* erfolgt innerhalb des Scriptprogrammes zum Speichern der Einstellungen, das wie folgt geändert wird (vergleiche auch Teil 12 dieser Reihe):

```

01 !Szene abspeichern
02 !-----
03 string itemID;
04 string i = "VSzene" # dom.GetObject("SzeneSet").State().Trunc().ToString();
05 i = i.Substr(0,i.Find("."));
06 dom.GetObject(i).State("");
07 var myAssembly = dom.GetObject(dom.GetObject("AktSzeneGewerk").State());
08 foreach(itemID, myAssembly.EnumUsedIDs())
09 {
10     var item = dom.GetObject(itemID);
11     if (item.IsTypeOf(OT_CHANNEL))
12     {
13         var device = dom.GetObject(item.Device());
14         if ((device.HssType().Find("HM-LC-Sw") >= 0))
15         {
16             if (dom.GetObject(i).State() == "")
17             {
18                 dom.GetObject(i).State(itemID # "*" # item.State());
19             }
20         }
21     }
22     dom.GetObject(i).State(dom.GetObject(i).State() # "," # itemID # "*" # item.State() );
23 }
24 }
25 if ((device.HssType().Find("HM-LC-Dim") >= 0))
26 {
27     if (dom.GetObject(i).State() == "")
28     {
29         dom.GetObject(i).State(itemID # "*" # item.State());
30     }

```

Geänderte Zeile 7:

```

var myAssembly =
dom.GetObject (dom.GetObject ("AktSzeneGewerk")
.State());

```



31	else
32	{
33	dom.GetObject(i).State(dom.GetObject(i).State() # "," # itemID # "*" # item.State());
34	}
35	
36	}
37	}
38	}
39	dom.GetObject("AktSzeneGewerk").State("");
40	dom.GetObject("SzeneSet").State(0);
41	

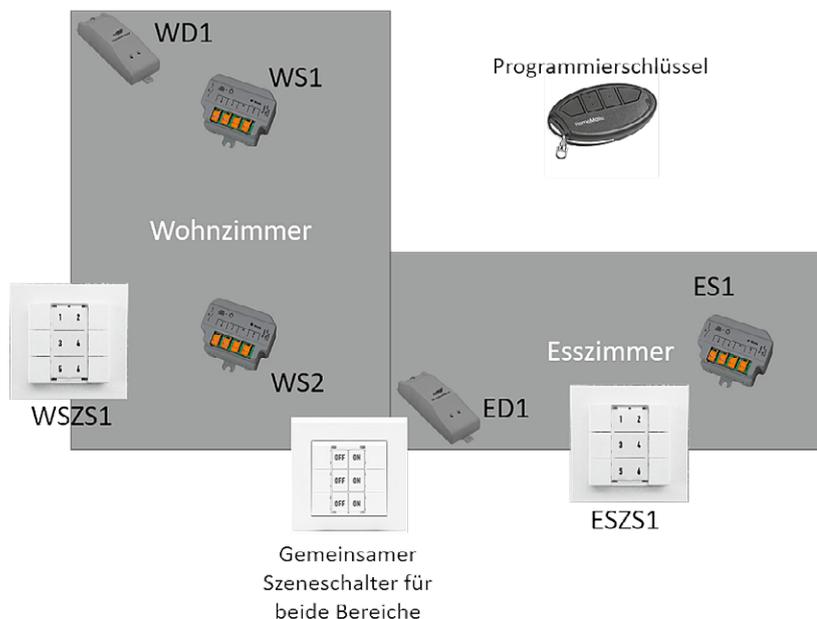
Neue Zeile 39:

⇒ Initialisierung der Systemvariablen (Zentralenvariablen) **AktSzeneGewerk**

Damit werden zum Anlegen eines neuen Szeneschalters folgende Schritte durchgeführt:

- (Neuen) Szeneschalter anlernen.
- Zentralenvariable VSzeneX (Typ Text) für jede Schaltfunktion des Schaltelementes erstellen (Datenspeicher für die Werte). X wird fortlaufend durchnummeriert. Die Zuordnung zwischen Nummerierung und Taster wird im neu zu erstellenden Zentralenprogramm (Schalterprogramm) festgelegt.
- Zentralenprogramm (Schalterprogramm) erstellen (erweitern).
- Gewerk für diesen Szeneschalter anlegen.
- Diesem Gewerk die zugehörigen Aktoren zuordnen.

Um das System nochmals zu verdeutlichen, hier ein Beispiel mit Homematic Komponenten (in diesem Beispiel sieht man sehr gut, wie flexibel das System ist und wie sich z. B. auch überlappende Bereiche schaffen lassen):



Wir haben folgende Konfiguration:

Den Raum Wohnzimmer mit den Aktoren

- Dimmer (WD1)
- Schalter (WS1)
- Schalter (WS2)

Den Raum Esszimmer mit den Aktoren

- Dimmer (ED1)
- Schalter (ES1)

Zur Bedienung gibt es folgende Szeneschalter:

- im Wohnzimmer den Szeneschalter WSZS1 (6 Tasten)
- im Esszimmer den Szeneschalter ESZS1 (6 Tasten)
- zentral (im „gemeinsamen Bereich“) den gemeinsamen Szeneschalter für beide Bereiche sowie
- einen „Programmierschlüssel“

Der Programmierschlüssel dient – wie in Folge 12 beschrieben – dazu, den Programmiermodus durch Tastendruck zu aktivieren. Dieser Modus bleibt eine kurze Zeit bestehen, in dieser Zeit kann man dann die aktuelle Einstellung einer Szene einlernen. Nach dieser Zeit wird der Programmiermodus wieder zurückgesetzt und die Tasten der Szeneschalter dienen nicht mehr zum Programmieren, sondern lediglich zum Abrufen der Szenen.



Es gibt drei Gewerke mit den Namen *SzeneGewerk1* (Wohnzimmer), *SzeneGewerk2* (Esszimmer) und *SzeneGewerk3* (gemeinsam für beide Räume).

Es gibt zwölf bis 18 Systemvariablen (Zentralenvariablen) vom Typ Text mit den Namen VSzene1 ... VSzene12 (VSzene18).

Des Weiteren gibt es drei Zentralenprogramme, in denen für die drei Taster die Variablen *SzeneON*, *SzeneSet* und *AktSzeneGewerk* bei einem Tastendruck gesetzt werden (siehe auch Folge 12 dieser Reihe).

Die Angabe zwölf bis 18 Systemvariablen rührt daher, dass unterschiedliche Zusammenstellungen beim bereichsübergreifenden Schalter auftreten können. Die Szeneschalter können beispielsweise folgendermaßen belegt sein:

	Szeneschalter Wohnzimmer						Szeneschalter Esszimmer						Gemeinsamer Szeneschalter beide Bereiche					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
VSzene1	X													X				
VSzene2		X																
VSzene3			X												X			
VSzene4				X														
VSzene5					X													
VSzene6						X									X			
VSzene7							X											
VSzene8								X								X		
VSzene9									X								X	
VSzene10										X								
VSzene11											X							
VSzene12												X						X
VSzene13													X					
VSzene14														X				
VSzene15															X			
VSzene16																X		
VSzene17																	X	
VSzene18																		X

Im ersten Beispiel (rot und schwarz) werden zwölf Datenspeicher (VSzene1 ... VSzene12) verwendet. Der gemeinsame Szeneschalter verwendet gleiche Datenspeicher wie die beiden anderen Szeneschalter.

Im zweiten Beispiel (rot und grün) werden für den gemeinsamen Schalter komplett getrennte Datenspeicher verwendet, sodass in Summe 18 benötigt werden.

Zwischen diesen beiden Beispielen sind alle Varianten möglich.

Die Zuordnung zu den Gewerken ist im Beispiel:

	AktSzeneGewerk1	AktSzeneGewerk2	AktSzeneGewerk3
Dimmer Wohnzimmer	X		X
Schalter 1 Wohnzimmer	X		X
Schalter 2 Wohnzimmer	X		X
Dimmer Esszimmer		X	X
Schalter 1 Esszimmer		X	X

Ausblick

Im nächsten und gleichzeitig letzten Artikel dieser Reihe werden wir nochmals eine Erweiterung/Vereinfachung der Scripte für die Szenenprogrammierung vorstellen. Wir werden ein Resümee ziehen und einen Ausblick auf die demnächst erscheinende Folgeserie geben, die sich mit der neuen CCU3 und dem Mediola Interface beschäftigt. 

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>