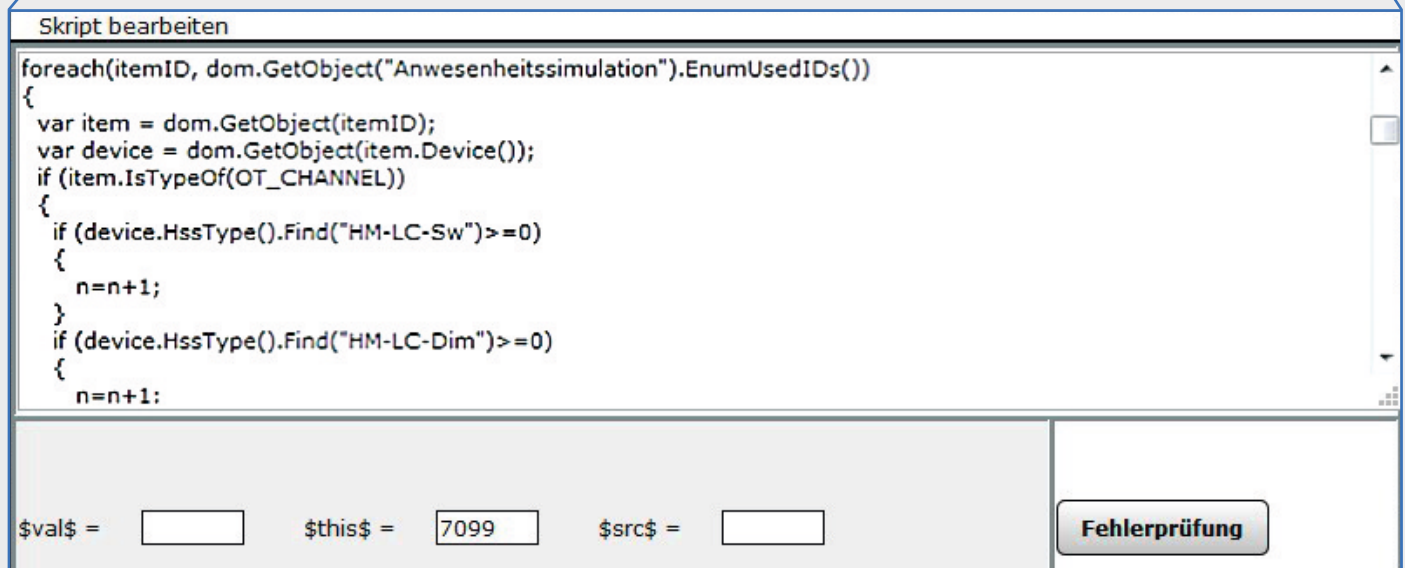
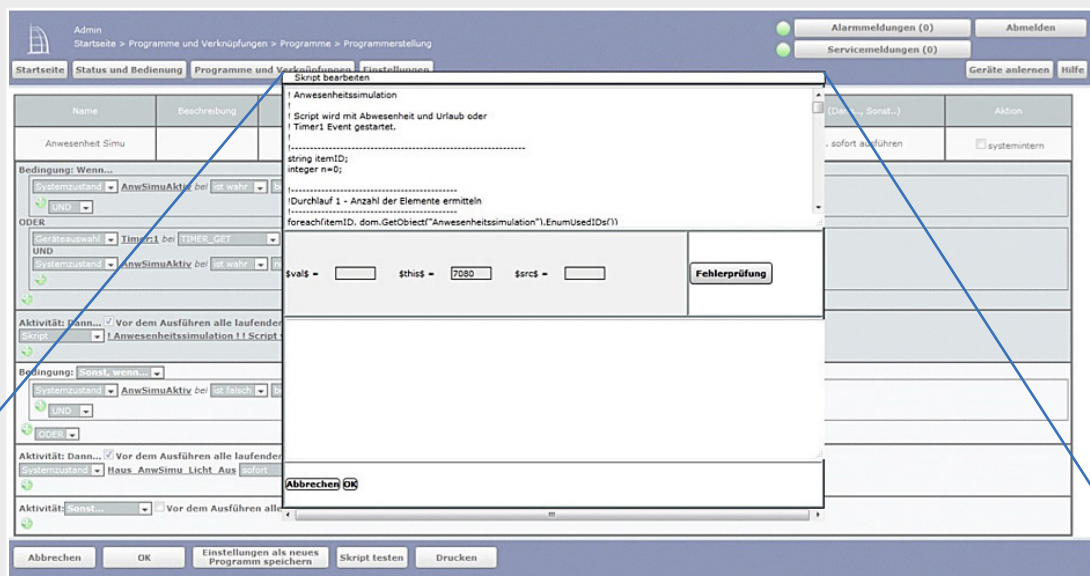




Homematic Scriptprogrammierung

Teil 9 – das mediola Interface und ein Texteingabescript



Im neunten Teil der Artikelserie beschäftigen wir uns weiter mit der Scriptprogrammierung im Zusammenhang mit dem mediola Gateway und dem mediola CREATOR und wir schreiben neue Scripte.

**Die Systemvariable (Zentralenvariable) *Eingabetext*:**

Die Zeichenkette, an der gearbeitet wird (der komplette Eingabestring), steht in der Zentralenvariablen *Eingabetext*.

Die Systemvariable (Zentralenvariable) *Eingabewert*:

Der Eingabewert entspricht der ASCII-Zahl des eingegebenen Zeichens (siehe ASCII-Tabelle) und wird durch das Drücken einer Taste auf einem virtuellen Keyboard auf der Eingabeseite des mediola Panels (s. u.) jeweils auf den Wert gesetzt.

Sobald sich dieser Wert ändert, wird das Script in einem Homematic Zentralenprogramm gestartet:

Name	Beschreibung	Bedingung (Wenn...)	Aktivität
Eingabe ASCII Mediola		Systemzustand: Eingabewert im Wertebereich größer als 0.00 bei Änderung auslösen	Systemzustand: Asciitabelle~,,0,1,2,3,4,5,6,7,8,9,,,,,,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P
Bedingung: Wenn...			
Systemzustand ☐ Eingabewert im Wertebereich größer als 0.00 bei Änderung auslösen ☐			
UND ODER			
Aktivität: Dann... ☒ Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).			
Systemzustand	Asciitabelle	sofort	~,,0,1,2,,
Skript	! ASC Eingabe dom.GetObject("Eingabetext").State(dom.GetObi...		sofort
Systemzustand	Eingabewert	sofort	0.00
Aktivität: Sonst... ☐ Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).			

Im ersten Schritt wird die Zentralenvariable *Asciitabelle* mit der Zeichenliste gefüllt.

In der Folge wird das Script zur Auswertung aufgerufen:

01	! ASC Eingabe
02	
03	dom.GetObject("Eingabetext").State(dom.GetObject("Eingabetext").State() () #dom.GetObject("Asciitabelle").State().StrValueByIndex(", ", dom.GetObject("Eingabewert").State()));

In der Programmzeile 03 sind wieder mehrere Zeilen zusammengefasst:

```
03a var Eingabezeichen = dom.GetObject("Eingabewert").State();
03b var BisherigerText = dom.GetObject("Eingabetext").State();
03c var AsciiListe = dom.GetObject("Asciitabelle").State();
03d var NeuerText= AsciiListe.StrValueByIndex(", ",Eingabezeichen);
03e dom.GetObject("Eingabetext").State(BisherigerText # NeuerText);
```

Ersetzen Sie einmal Zug um Zug die (im Script lokalen) Variablen *BisherigerText* sowie *NeuerText* durch die in den Zeilen 03a bis 03e von den Gleichheitszeichen passenden rechten Seiten. Das Ergebnis wird die Zeile 3 sein.

→ Mit dieser Zeile 3 wird also die vom mediola Interface übergebene Zahl in den korrespondierenden Buchstaben umgewandelt, indem dieser in der Auflistung in *Asciitabelle* geholt und dann an den bisherigen Text angehängt wird.

Nun fehlt uns noch die Möglichkeit, das zuletzt eingegebene Zeichen zu löschen:

04	! Delete (rueckwaertsloeschen)
05	if (dom.GetObject("Eingabewert").State()==127){
06	dom.GetObject("Eingabetext").State(dom.GetObject("Eingabetext").State() ().Substr(0,dom.GetObject("Eingabetext").State().Length()-1));
07	}

Auch hier stecken wieder mehrere Programmzeilen dahinter:

```
06a var Eingabezeichen = dom.GetObject("Eingabewert").State();
06b var BisherigerText = dom.GetObject("Eingabetext").State();
06c var LaengeBisherigerText = BisherigerText.Length()
06d var GekuerzterText = BisherigerText.Substr(0,LaengeBisherigerText-1);
06e if (Eingabezeichen==127){
06e     dom.GetObject("Eingabetext").State(GekuerzterText);
06f }
```



Da das Komma in der Liste (Asciitabelle) als Listenelement nicht erlaubt ist, weil es in der Textvariablen der Listentrenner ist, wird es separat behandelt:

08	!Komma einsetzen
09	!-----
10	if (dom.GetObject("Eingabewert").State()==44){dom.GetObject("Eingabetext").State(dom.GetObject("Eingabetext").State()#",");}

Die ausführliche Schreibweise der Zeile 10:

```
10a var Eingabezeichen = dom.GetObject("Eingabewert").State();
10b var BisherigerText = dom.GetObject("Eingabetext").State();
10c if (Eingabezeichen == 44){
10d     dom.GetObject("Eingabetext").State(BisherigerText # ",");
10e }
```

Die Umlaute befinden sich in der ASCII-Tabelle im hinteren Bereich, entweder trägt man sie in die Definitionszeile für *Asciitabelle* nach einer entsprechenden Anzahl von Kommata noch ein oder man behandelt sie entsprechend dem Komma separat;

11	! ß einfüegen
12	if (dom.GetObject("Eingabewert").State()==128){dom.GetObject("Eingabetext").State(dom.GetObject("Eingabetext").State()#"ß");}
13	! ü einfüegen
14	if (dom.GetObject("Eingabewert").State()==129){dom.GetObject("Eingabetext").State(dom.GetObject("Eingabetext").State()#"ü");}
15	! ö einfüegen
16	if (dom.GetObject("Eingabewert").State()==130){dom.GetObject("Eingabetext").State(dom.GetObject("Eingabetext").State()#"ö");}
17	! ä einfüegen
18	if (dom.GetObject("Eingabewert").State()==131){dom.GetObject("Eingabetext").State(dom.GetObject("Eingabetext").State()#"ä");}
19	! Ü einfüegen
20	if (dom.GetObject("Eingabewert").State()==132){dom.GetObject("Eingabetext").State(dom.GetObject("Eingabetext").State()#"Ü");}
21	! Ö einfüegen
22	if (dom.GetObject("Eingabewert").State()==133){dom.GetObject("Eingabetext").State(dom.GetObject("Eingabetext").State()#"Ö");}
23	! Ä einfüegen
24	if (dom.GetObject("Eingabewert").State()==134){dom.GetObject("Eingabetext").State(dom.GetObject("Eingabetext").State()#"Ä");}

Die Systemvariable (Zentralenvariable) *ASCII_INPUT_SELECT*:

Wenn im mediola Interface die Taste ENTER gedrückt wird, wird die Zentralenvariable *ASCII_INPUT_SELECT* mit einem Wert beschrieben, der in unserem Homematic Script bestimmt, wohin der eingegebene Text geschrieben wird (in welche Systemvariable/Zentralenvariable). Damit lässt/lassen sich das/die Eingabefenster (s. u.) universell nutzen und muss/müssen nur einmal entworfen werden.

25	! ASCII_INPUT_SELECT = 1001 - Variable Asciitabelle
26	if (dom.GetObject("Eingabewert").State()==1001)
27	{
28	dom.GetObject("Asciitabelle").State(dom.GetObject("Eingabetext").State());
29	}

30	! ASCII_INPUT_SELECT = 1002 - Variable WasAuchImmer01
31	if (dom.GetObject("Eingabewert").State()==1001)
32	{
33	dom.GetObject("WasAuchImmer01").State(dom.GetObject("Eingabetext").State());
34	}



35	! ASCII_INPUT_SELECT = 1003 - Variable WasAuchImmer02
36	if (dom.GetObject("Eingabewert").State()==1001)
37	{
38	dom.GetObject("WasAuchImmer02").State(dom.GetObject("Eingabetext").State());
39	}

USW.

Auch hier noch einmal der Hinweis: Die letzten 3 Blöcke des Scripts (Zeilen 25...29, 30...34 und 35...39) lassen sich auch allgemein gültig und damit vereinfacht programmieren, wenn die Zuordnungsvorschrift (Codenummer 1001, 1002, 1003 ... zu Variablenname *Asciitabelle*, *WasAuchImmer01*, *WasAuchImmer02* ...) ebenfalls in eine Variable (Liste) geschrieben wird:

Zentralenvariable:

```
ASCII_Zuordnung = "1001(Asciitabelle),1002(WasAuchImmer01),1003(WasAuchImmer02) "
```

Damit sieht das Script, das die 3 Blöcke ersetzt, folgendermaßen aus:

25	! Zieldecodierung 1001, 1002, 1003, ... , 1500
26	if (dom.GetObject("Eingabewert").State()==1500)
27	{
28	var daten = dom.GetObject("ASCII_Zuordnung").State();
29	var position1= daten.Find(dom.GetObject("ASCII_INPUT_SELECT").State().ToString().Substr(0,4));
30	if (position1>=0)
31	{
32	var daten=daten.Substr(position1+5,daten.Length()-position1);
33	var Variablenname=daten.Substr(0,daten.Find(" "));
34	dom.GetObject(Variablenname).State(dom.GetObject("Eingabetext").State());
35	}
36	}
37	dom.GetObject("ASCII_INPUT_SELECT").State(0);

Der Scriptteil wird nur abgearbeitet, wenn die Systemvariable *Eingabewert* den Wert 1500 annimmt.

Im Beispiel soll der Wert in der Systemvariable *ASCII_INPUT_SELECT* = 1002 sein:

Im Script wird zunächst die Zuordnung Nummer zu Variablenname (*ASCII_Zuordnung*) in eine Scriptvariable (*daten*) eingelesen (Zeile 28).

Im nächsten Schritt wird der String *daten* auf das Vorhandensein der Zahl überprüft, die in der Systemvariablen *Eingabewert* übergeben wird (Zeile 29). Ist die Zahl nicht vorhanden, nimmt die Scriptvariable *position1* den Wert -1 an, ansonsten die Position im String *daten* (die erste Position ist die 0!). Dies ist in unserem Beispiel die 19.

Nur, wenn die Zahl gefunden wurde, wird der folgende Scriptteil abgearbeitet (Zeile 30 und die folgenden Zeilen).

Zunächst wird vom Gesamtstring

```
(in unserem Beispiel: "1001(Asciitabelle),1002(WasAuchImmer01),1003(WasAuchImmer02) ")
```

der vor der fünften Position nach der gefundenen Position (Position 19+5 = 24) befindliche Teil des Strings abgetrennt (Zeile 32) und ergibt damit für *daten*:

```
"WasAuchImmer01),1003(WasAuchImmer02) "
```

Danach wird der hintere Teil ab dem Zeichen „)" abgetrennt (Zeile 33) und das Ergebnis in die lokale Scriptvariable Variablenname gespeichert.

```
Variablenname = "WasAuchImmer01"
```

Dies ist dann auch der gesuchte Variablenname, der zur Nummer 1002 gehört und diese (System-)Variable wird dann mit dem Eingabetext beschrieben (Zeile 34).

Zum Schluss wird die Systemvariable (Zentralenvariable) *ASCII_INPUT_SELECT* = 0 gesetzt.

Damit kann durch Hinzufügen einer Systemvariablen ohne Programm-(Script-)änderung – lediglich durch Erweitern der Systemvariable *Asciitabelle* – erweitert werden.



Die „Mediola Seite“:

Zum Abschluss noch zur mediola Seite, die uns ja die Eingabeseite(n) für die Texte zur Verfügung stellen soll. Es wird eine Seite entworfen, die die Form eines virtuellen Keyboards hat:



Die einzelnen Tasten setzen bei einem Druck die Systemvariable *Eingabewert* (CCU-Zentralenvariable) auf den ASCII-Wert der Tabelle (s. o.). Beispiel: A → 65, B → 66, C → 67 ...

Wenn die Eingabeseite aufgerufen wird, dann wird die CCU-Systemvariable *ASCII_INPUT_SELECT* mit der Codenummer derjenigen Variablen beschrieben, die beim Verlassen der Eingabeseite die Systemvariable kennzeichnet, die mit dem Eingabetext beschrieben werden soll (1001 ... 1499).

Dies geschieht durch eine sogenannte Szene in der mediola Oberfläche, die zwei Befehle beinhaltet, wobei der erste die Systemvariable *ASCII_INPUT_SELECT* setzt und der zweite die Keyboardseite aufruft.

Beim Drücken der ENTER-Taste auf der Keyboardseite wird der Eingabewert auf 1500 gesetzt, was im oben beschriebenen Script dafür sorgt, dass der Wert in *ASCII_INPUT_SELECT* zur Suche der Systemvariablen verwendet wird, in die der eingegebene Text geschrieben werden soll.

Die Rücklöschtaaste wird mit der Zahl 127 belegt.

Eine zweite Keyboardseite wird aufgerufen, wenn die Shift-Taste gedrückt wird (/ \). Die Seite ist für die Eingabe der Shift-Tastenbelegungen vorgesehen. Die ENTER-Taste auf dieser Seite hat die gleiche Funktion wie die der Eingabeseite mit den Non-Shift-Tastenbelegungen:



Der nächste Artikel beschäftigt sich mit Scripten rund um die Sicherheit im Haus. **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 Unternehmensegruppe 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 Internet-Plattformen 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>